ORSAY

N° d'ordre : 2542

THESE

PRESENTEE

A L'UNIVERSITE DE PARIS SUD
CENTRE D'ORSAY

POUR OBTENIR

LE GRADE DE DOCTEUR ES SCIENCES PHYSIQUES

par

Mayez AL MOUHAMED
Docteur de troisième cycle

CONTRIBUTION A LA COMMANDE AUTOMATIQUE EN ROBOTIQUE

APPLICATION A LA COMMANDE PAR MICROPROCESSEURS DES MANIPULATEURS ARTICULES

Soutenue le 29 janvier 1982, devant la Commission d'Examen

MM. P. DEBRAINE — Président
P. COIFFET
L. DONADIEU
A. BONNEMAY
B. de COSNAC
J. VERTUT

ORSAY

N° d'ordre : 2542

THESE

PRESENTEE

A L'UNIVERSITE DE PARIS SUD

CENTRE D'ORSAY

POUR OBTENIR

LE GRADE DE DOCTEUR ES SCIENCES PHYSIQUES

par

Mayez AL MOUHAMED

Docteur de troisième cycle

CONTRIBUTION A LA COMMANDE AUTOMATIQUE EN ROBOTIQUE

APPLICATION A LA COMMANDE PAR MICROPROCESSEURS DES MANIPULATEURS ARTICULES

Soutenue le 29 janvier 1982, devant la Commission d'Examen

MM. P. DEBRAINE — Président

P. COIFFET

L. DONADIEU

A. BONNEMAY

B. de COSNAC

J. VERTUT

UNIV.
LEIDEN
BIBL.

REMERCIEMENTS

Je tiens à exprimer ma profonde reconnaissance à M. P. DEBRAINE, Professeur à l'Institut National des Sciences et Techniques Nucléaires, qui m'a fait l'honneur de diriger cette thèse.

Que M. P. COIFFET, Chargé de la Robotique au Laboratoire d'Automatique de Montpellier, trouve ici l'expression de ma profonde gratitude.

Je remercie M. le Professeur L. DONADIEU et M. A BONNEMAY pour avoir bien voulu faire partie de mon jury.

Je remercie tout particulièrement M. B. De COSNAC, Adjoint au Chef du Service d'Instrumentation pour la Recherche, pour l'enseignement de valeur et les conseils qui ont constamment guidé cette recherche.

Je remercie vivement M. J. VERTUT, Chef du Service des Techniques d'Equipement de Protection et de Dosimétrie, pour ses conseils prodigués et pour les robots qu'il a mis à ma disposition.

J'exprime ma reconnaissance à M. J. WEILL, Chef du Département d'Electronique et d'Instrumentation Nucléaire, et à M. Y. AMRAM, Chef du Service d'Instrumentation pour la Recherche, qui m'ont assuré les meilleures conditions de recherche possibles.

Que MM. A. BASIRE, E. CSAKVARY, J.M. MARTINEZ, H. AL MOHAMAD et D. JUVIN trouvent ici l'expression de ma profonde gratitude pour leur apport à cette recherche et leur amitié.

TABLE DES MATIERES

INTRODUCTION

CHAPITRE I - COORDINATION DES MOUVEMENTS

I.1 - Introduction

I.2 - Modèle géométrique d'une chaîne articulée

I.3 - Représentation de l'effecteur

I.3.1 - Diverses représentations de l'effecteur

I.3.2 - Représentation utilisant une variante des angles d'Euler

I.3.2.1 - Définition du vecteur de l'effecteur

I.4 - Inversion du modèle géométrique

I.4.1 - Problématique du modèle géométrique inverse

I.4.2 - Unicité et singularité

I.4.3 - Remarques et conclusion

I.5 - Modèle variationnel et problème de l'inverse

I.5.1 - Introduction

I.5.2 - Modèle cinématique et variationnel

I.5.3 - Inversion du modèle variationnel

I.5.4 - Conclusion

I.6 - Synthèse d'un système de coordination général

I.6.1 - Définition d'un système de coordination des mouvements

I.6.2 - Principaux modes de commande

I.6.3 - Application à la coordination pour un robot

I.7 - Conclusion.

CHAPITRE II - COORDINATION EN EFFORT

II.1 - Introduction

II.2 - Compliance et centre d'élasticité

II.3 - Comportement et stabilité de la régulation en effort
II.4 - Calcul du vecteur des forces généralisées à partir des couples moteurs
II.5 - Obtention de l'information à partir d'un capteur d'effort
II.6 - Synthèse de fonctions pour la régulation de l'effort
II.7 - Application aux fonctions de manipulation et d'assemblage
II.8 - Conclusion.

CHAPITRE III - COMMANDE DYNAMIQUE DES SYSTEMES ARTICULES

III.1 - Introduction
III.2 - Equation de comportement d'un mécanisme articulé
III.2.1 - Représentation d'un solide
III.2.2 - Equation de mouvement
III.2.3 - Conclusion
III.3 - Obtention directe des paramètres de l'équation de comportement
III.3.1 - Modélisation de la chaîne organe moteur - réducteurs - articulation
III.3.2 - Principe de l'identification et ordonnancement
III.3.3 - Lissage d'une fonction multivariable
III.3.4 - Application au manipulateur M.A.23
III.5 - Conclusion
III.4 - Calcul en ligne des paramètres de la dynamique
IV.4.1 - Optimisation avec la représentation analytique des paramètres
IV.4.2 - Optimisation avec la représentation discrète des paramètres
III.5 - Procédés de contrôle en dynamique
III.5.1 - Découplage au niveau des variables généralisées
III.5.2 - Découplage au niveau du vecteur de l'effecteur
III.6 - Application à la commande dynamique d'un manipulateur

III.6.1 - Amélioration des performances

III.7 - Conclusion.

CONCLUSION GENERALE

BIBLIOGRAPHIE

ANNEXES :

A1 - Modèle géométrique inverse pour le manipulateur M.A.23
A2 - Calcul du vecteur des forces
A3 - Obtention de modèles de transfert simples
A4 - Calcul des énergies cinétique et potentielle.
A5 - Extraits de programmes.

Digitized by the Internet Archive
in 2026

https://archive.org/details/IA41547318_0002

INTRODUCTION

Le caractère évolutif de la société humaine est le résultat d'une longue mutation marquée par une recherche permanente du renouveau par la méthode et le matériau-outil. La découverte de lois et de techniques nouvelles a permis toujours à l'homme, par sa fonction, de se situer plus haut dans l'appareil productif. La problématique de la mécanisation et de l'automatisation de cet appareil consiste à éviter de le changer complètement lors de l'apparition de techniques nouvelles, de la nécessité de produire autrement ou d'explorer des domaines nouveaux. On tente alors d'apporter une certaine généralité dans le système productif et ce par des considérations sur le système d'interface homme-machine et non pas directement au niveau de la machine elle-même.

Dans ce contexte la robotique apparaît comme une discipline permettant la mise en oeuvre des moyens pour prolonger ou transposer les actions motrices et intelligentes de l'homme. Dans un premier temps l'objectif visé consiste à trouver une méthode-robot pour remplacer l'homme dans certains domaines jugés dangereux (rayonnement, milieu sous-marin, travaux de fonderie, etc.) ou nuisibles (troubles psychomoteurs dans les travaux à caractère répétitif).

La robotique d'aujourd'hui est une science qui représente un champ d'application intéressant pour des disciplines diverses telles que la mécanique, l'électronique, et l'intelligence artificielle. Le progrès accompli dans ces diverses matières a permis de passer du stade du robot-asservissement à celui des robots dits évolués. Il s'agit de robots munis de sens leur permettant d'exécuter des tâches prévues dans leurs gros plans. Ils sont capables de prendre des décisions issues d'une logique appliquée aux informations délivrées par des senseurs d'environnement.

Les machines-outils sont conçues pour exécuter des tâches précises dans des conditions prévues dans leurs moindres détails. Bien qu'elles soient très efficaces dans leurs domaines, elles ne tolèrent pas de modification de leurs conditions opératoires. Le robot évolué permet une adaptation à des tâches et des objets différents. La versatilité d'une machine-robot se traduit par sa capacité de s'adapter aux variations des tâches et des objets manipulés. Il s'agit d'une condition impérative à l'entrée de ces machines au stade industriel pour lequel le robot doit s'adapter aux normes variables de la production.

Vu sous l'aspect d'un mécanisme interprétatif le schéma du robot évolué comprend la compréhension du message ou de l'ordre et la possibilité d'y répondre par interaction avec l'environnement. Par leur nature ces robots s'accomodent de langages descriptifs permettant des spécifications fonctionnelles de la tâche et du robot lui-même. Le problème d'un langage commun à tous les robots reste de nos jours un domaine extrêmement difficile et entièrement ouvert. La façon dont le message doit être interprété dépend, pour une large part, des capacités interactives du robot. C'est ainsi que les recherches entreprises s'orientent vers la structuration des tâches, appartenant à un même niveau d'interactivité, dans le but d'en faire des éléments de la commande.

D'une manière générale, le message, donné au robot, conduit à saisir l'objet-outil, le transporter au lieu présumé de la tâche et agir sur lui d'une manière convenable pour la bonne exécution de cette dernière. Cette scène d'action se décompose en éléments plus simples qui sont très variés. La prise de l'objet s'effectue essentiellement au moyen d'informations visuelles. Mais la manière avec laquelle le robot saisit l'objet dépend de l'opération que va subir ce dernier et des possibilités du robot lui-même. Le transport de l'objet relève du problème de la coordination des mouvements et de l'élaboration de trajectoires. Ces dernières ne doivent pas entraîner de collision entre la partie en mouvement, le robot et l'objet, et le reste de l'environnement. Cette façon d'éviter les obstacles peut se

faire sur la base d'une modélisation de l'univers du robot ou de manière interactive où les capteurs de proximité fournissent des informations permettant de modifier ces trajectoires. La tâche proprement dite est la partie la plus difficile car elle fait appel à plusieurs sens à la fois tels que le sens du toucher ou l'effort et la vision. En effet, la précision intrinsèque du robot est largement insuffisante dans la plupart des tâches industrielles. Le processus d'affinement s'effectue par le moyen de capteurs (capteurs visuels, de forces, de proximité, de non homogénéité de métaux, etc.) qui prennent alors un sens important en robotique. Au cours de l'exécution de la tâche l'écart entre la situation idéale (idée transmise par l'homme) et la situation réelle (connue par les capteurs) est décomposé et converti en actions correctives élémentaires dont la conséquence doit entraîner, a priori, une avance vers l'exécution de la tâche.

Le robot n'est compétitif, sur le plan économique, que si sa cadence de production approche celle de l'homme. Lorsque la machine-robot fonctionne à des vitesses élevées (quelques mètres par seconde) elle est le siège de forces dynamiques conduisant à des écarts intolérables vis-à-vis des trajectoires prescrites. Les forces de gravité réduisent de façon sensible la précision intrinsèque du robot notamment lorsque ce dernier est en charge (de quelques unités à quelques dizaines de kg). Le contrôle dynamique d'un robot conduit à compenser ces perturbations et par conséquent à aller plus vite avec une meilleure stabilité.

Nous nous sommes intéressés, de façon générale, à l'étude du comportement adaptatif des robots manipulateurs. Notre démarche traite le problème de l'analyse et de la synthèse des éléments avec lesquels ce comportement peut être obtenu. Nous avons choisi de répartir notre exposé sur trois thèmes généraux : la coordination des mouvements, la coordination des efforts et le contrôle dynamique des robots manipulateurs.

Le premier thème consiste en l'étude et la synthèse d'un système général de coordination des mouvements. Le robot en tant que système articulé n'est accessible à l'utilisateur que lorsque ses paramètres de positionnement globaux sont liés à sa partie préhensive. Nous étudions les moyens d'associer, à une telle représentation du robot, les variables articulaires spécifiant la configuration de ce dernier. Nous montrons en particulier comment des heuristiques simples peuvent rendre quasiment biunivoques les relations du changement de coordonnées.

Dans les systèmes traditionnels le point du robot sur lequel on agit (centre d'action) se situe sur le segment terminal en particulier au centre de prise. Si cette représentation convient dans la phase de transport de l'objet elle ne l'est pas dans les opérations d'assemblage où les corrections doivent s'effectuer par rapport à un point quelconque de l'objet manipulé. Nous avons généralisé la notion de centre d'action afin de pouvoir agir par rapport à un point situé à l'extérieur du robot en particulier à un endroit de l'objet. L'ordre donné au robot conduit alors à agir sur le repère propre de l'objet manipulé. Ces fonctions constituent les éléments correctifs indispensables pour les mécanismes rebouclés.

En application nous définissons un système de coordination construit avec cette approche sur une unité à microprocesseur 16 bits (INTEL 8086). Le fonctionnement de ce système est illustré par des courbes d'évolution. Le cycle de coordination dure 8 ms contre une durée de 30 ms sur un mini-calculateur (nous utilisons un coprocesseur flottant INTEL 8087).

Le second thème traite le problème de l'information d'effort sous l'aspect de la conception de capteurs de forces généraux. Les capteurs spécialisés ou capteurs-outils sont orientés vers des tâches et des conditions précises. Nous présentons un capteur de forces (une pince) comprenant trois degrés de liberté. Ce dernier recueille des informations sur les forces extérieures que subit l'objet au cours de l'assemblage. Ces in-

formations serviront pour obtenir une compliance* active très utile dans les opérations d'assemblage. Peu de systèmes, à l'heure actuelle, comprennent des fonctions de régulation en effort. Nous présentons l'analyse et la synthèse de ces mécanismes permettant de réguler la force qu'exerce le robot par rapport à celle recueillie sur le capteur. Leur caractéristique est de permettre, dans les opérations d'assemblage, des corrections de position et d'orientation vis-à-vis du centre d'élasticité qui est alors confondu, par programmation, avec le centre d'action.

En application nous utilisons ces mécanismes de régulation pour engendrer les fonctions adaptatives telles que POSER, INSERER et EXTRAIRE.

Le troisième thème nous conduit à l'analyse du comportement dynamique des systèmes articulés. Après avoir établi l'équation de mouvement, par le moyen d'un modèle mathématique, nous analysons les différentes forces dynamiques et statiques qui y sont présentes. Ces forces provoquent des écartements importants, par rapport aux trajectoires de consignes, lorsque la vitesse et/ou la charge sont élevés. Une commande plus performante (le contrôle traditionnel s'effectue avec des régulations dites proportionnelles et dérivées) consiste à compenser ces forces au cours du mouvement afin de mieux contrôler l'évolution réelle du système articulé. Mais la caractéristique importante de ces forces est d'apparaître et de disparaître vite. Par conséquent la qualité d'une compensation dépend, dans une large part, de la précision et de la rapidité du calcul de ces forces. L'examen des expressions de ces dernières (obtenues d'après le modèle mathématique) montre une grande complexité. De plus ce modèle ne permet pas de mettre en évidence des expressions simplifiées de ces forces qui restent valables lors des variations de configuration, de vitesse et de la charge. Cela ex-

---

* Compliance, c'est la possibilité pour une structure de se déformer afin de minimiser les forces et les couples extérieurs auxquels elle est soumise

plique, en partie, les difficultés que rencontre une commande avec le modèle notamment lorsque les variables articulaires interviennent avec le même poids dans la procédure de calcul en temps réel (difficulté pour choisir les variables prépondérantes).

La méthode que nous proposons consiste à identifier, sur le système réel, les coefficients intervenant dans les expressions de ces forces. Cela revient à procéder à un apprentissage du comportement dynamique pour un système articulé. Notre démarche tente de réduire de façon notable, dans le calcul des forces de compensation, l'influence des erreurs de modélisation d'une structure électro-mécanique. Elle permet en outre de suivre l'évolution de ces coefficients au cours du vieillissement du matériau mécanique.

Sur le plan fonctionnel cette méthode revient à appliquer un couple d'excitation sur une articulation (pour étudier le couplage entre cette dernière et le reste du système) et conduit en général à des évolutions angulaires sur les différentes articulations du système. Pour dégager ces coefficients (coefficients d'inertie) nous utilisons des modèles simples pour décrire l'interaction entre l'organe moteur et l'articulation (la modélisation d'une chaîne moteurs-réducteurs-articulations a été menée au L.A.M.). Une analyse fine permet alors d'estimer ces coefficients pour une configuration articulaire donnée. Ainsi l'analyse des effets réels, qui se manifestent clairement dans certains mouvements, permet l'obtention d'une représentation discrète de coefficients dynamiques.

La représentation discrète des coefficients offre deux possibilités principales pour leur exploitation dans la loi de commande. La première consiste à utiliser les techniques de tabulation et d'interpolation multivariable. La seconde consiste à leur associer une représentation polynominale en fonction des variables articulaires.

En application nous montrons la structure de notre commande dynamique et sa liaison avec le système de coordination des mouvements. Nous présentons des courbes d'évolution réelle du robot avec et sans la correction dynamique (amélioration des performances). Ce système est implanté sur notre unité à microprocesseur qui requiert 9 ms pour le calcul des forces dynamiques de compensation. Nous présentons également une structure d'asservissement analogique particulièrement adaptée à la commande dynamique des systèmes articulés.

Ce mémoire est présenté avec un ensemble de démonstrations filmées portant sur la coordination des mouvements, les fonctions adaptatives (INSERER, EXTRAIRE, POSER et PRENDRE) et la commande dynamique (transport d'une charge).

CHAPITRE I

COORDINATION DES MOUVEMENTS

I.1 - INTRODUCTION

Notre objectif, dans ce chapitre, est d'étudier une structure de coordination comportant les moyens d'agir, sur un robot, conformément aux contraintes liées à la spécificité des tâches. L'action du robot, sur un outil ayant certains degrés de liberté par rapport à un support fixe, ne peut s'accomplir que si ce dernier peut coordonner ses mouvements par rapport à ces degrés de liberté. C'est pourquoi l'objectif de cette partie est de dégager les éléments d'une telle coordination et d'étudier la possibilité de sa mise en oeuvre à un niveau très bas dans la hiérarchie de la commande. Par sa fonctionnalité cette démarche situe la coordination à un niveau proche de celui de la machine mécanique. Elle soulève donc des problèmes spécifiques qui auraient des solutions plus simples lorsqu'ils sont traités sur des unités de calcul traditionnelles (les mini-calculateurs classiques).

La tâche demandée au robot est décrite en termes d'évolution dans le temps de son organe de préhension. Ce dernier, appelé également l'effecteur du robot, doit agir sur les objets qui l'entourent de façon réfléchie. Le problème de la représentation de cette partie du robot est déterminant car c'est le moyen pour l'homme de spécifier le geste qu'il demande à ce dernier. Nous traiterons cette question dans les paragraphes I.1 et I.2.

Par la nature articulaire du robot, ce dernier est asservi en position par rapport à ces articulations. D'où l'intérêt de faire correspondre à une évolution de l'effecteur une évolution des grandeurs articulaires. L'étude de cette

correspondance, ou changement de coordonnée, est traitée dans les paragraphes I.3 et I.4.

La structuration des formes d'action sur le robot répond à des objectifs de simplicité de dialogue et d'adaptation générale à la tâche. Nous présenterons le problème de la synthèse des fonctions de mouvements coordonnés au paragraphe I.5.

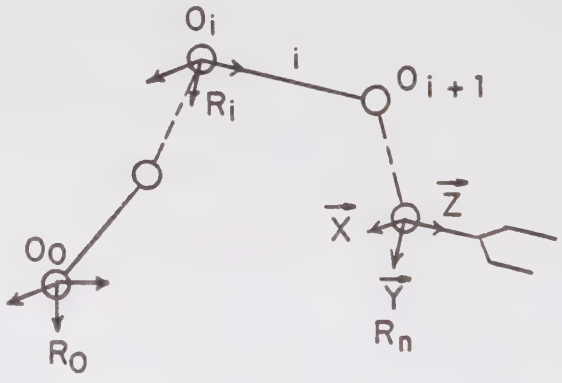
Nous présentons au paragraphe I.6 une application à la coordination pour un manipulateur à 6 degrés de liberté. Nous décrivons, brièvement, cette structure dans le cadre d'une implantation sur une unité à microprocesseur.

I.2 - MODELE GEOMETRIQUE D'UNE CHAINE ARTICULEE

Un mécanisme articulé est un système composé de n segments se mouvant les uns par rapport aux autres. Le segment n° i, de la chaîne, est repéré par rapport à ses points de liaison : O_{i-1} avec le segment précédent et O_i avec le segment suivant.

Par rapport à un référentiel fixe R_o le vecteur $\overrightarrow{O_oO_i}$ s'exprime comme la somme vectorielle d'un vecteur qui caractérise la chaîne amont au ième segment et du vecteur $\overrightarrow{O_{i-1}O_i}$:

$$\overrightarrow{O_oO_i} = \overrightarrow{O_oO_{i-1}} + \overrightarrow{O_{i-1}O_i}$$



Représentation d'un segment dans une chaîne articulée

Un référentiel R_i lié au ième degré de liberté est placé au point O_{i-1}. Les vecteurs orthonomés de R_i s'expriment, par rapport à R_o, par le produit de toutes les matrices de passages M_j (j : 1,..., i) qui lient deux segments consécutifs :

$$M_{i/R_o} = \left[\vec{X}_i, \vec{Y}_i, \vec{Z}_i \right] = \prod_{j=1}^{i} M_{j/R_o}$$

où $\vec{X}_i$, $\vec{Y}_i$, $\vec{Z}_i$ sont les vecteurs orthonomés de R_i.

Lorsque le degré de liberté associé au segment n° i est rotoïde le vecteur $\overrightarrow{O_{i-1}O_i}$ s'exprime de la façon suivante :

$$\overrightarrow{O_{i-1}O_i}_{/R_o} = M_{i/R_o} \cdot \overrightarrow{O_{i-1}O_i}_{/R_i}$$

Mais si ce même degré de liberté est prismatique * nous aurons :

$$\overrightarrow{O_{i-1}O_i}_{/R_o} = M_{i-1/R_o} \cdot \overrightarrow{O_{i-1}O_i}_{/R_{i-1}}$$

exprimant ainsi que le repère R_i reste, dans ce cas, parallèle à R_{i-1}. Le segment terminal de la chaîne est déterminé par le vecteur $\overrightarrow{O_oO_n}$ et la matrice M_{n/R_o} définissant respectivement sa position et son orientation par rapport au repère fixe R_o. Ces grandeurs se résument par les expressions suivantes :

$$\overrightarrow{O_oO_n}_{/R_o} = \left[x(\theta), y(\theta), z(\theta) \right]^t$$

et $$M_{n/R_o} = \left[\vec{X}(\theta), \vec{Y}(\theta), \vec{Z}(\theta) \right]$$

où - $\overrightarrow{O_oO_n}$ est un vecteur de dimension (3,1)

$\overrightarrow{M_{n/R_o}}$ est une matrice de dimension (3,3)

- $\theta = (\theta_1, \ldots, \theta_n)^t$ désigne le vecteur de configuration de la chaîne.
  La variable θ_i est la coordonnée généralisée du ième segment par rapport au segment qui le précède.

- $x(\theta)$, $y(\theta)$ et $z(\theta)$ sont les coordonnées cartésiennes du point O_n, ce sont des fonctions multivariables des θ_i

---

* Liaison permettant une translation d'une articulation par rapport à celle qui la précède

(i : 1,..., n)

- $\vec{X}(\theta)$, $\vec{Y}(\theta)$ et $\vec{Z}(\theta)$ sont les vecteurs orthonormés du repère R_n.

I.3 - REPRESENTATION DE L'EFFECTEUR

D'une manière générale le but de la commande est d'agir sur les objets qui se trouvent dans l'espace de travail du robot. La partie de ce dernier qui est appelée à s'attacher aux objets pour les manipuler est dite organe de préhension ou effecteur du robot. Ainsi la tâche demandée est décrite en termes d'évolution de l'effecteur en tant que vecteur à six dimensions. Nous avons vu dans le paragraphe précédent que l'effecteur est caractérisé par son vecteur origine $\overrightarrow{O_o O_n}(\theta)$ et par sa matrice de rotation généralisée $M_n(\theta)$. La recherche d'une représentation propre à l'effecteur vient du fait que la matrice $M_n(\theta)$ n'est pas commode à utiliser dans une représentation extérieure de l'orientation de ce dernier (par exemple dans la programmation des mouvements). Nous verrons dans ce paragraphe plusieurs types de représentations en particulier nous nous intéressons à celles utilisant une variante des angles d'Euler ainsi qu'aux problèmes qui en découlent.

I.3.1 - Diverses représentations de l'effecteur

Une façon simple, pour représenter l'effecteur, consiste à considérer les coordonnées cartésiennes du point terminal aux trois premiers degrés de liberté (appelé également origine du poignet) en leur associant pour l'orientation les degrés de liberté restants :

$$E = \{ x(\bar{\theta}), y(\bar{\theta}), z(\bar{\theta}), \theta_4, \theta_5, \theta_6 \} \qquad \bar{\theta} = (\theta_1, \theta_2, \theta_3)^t$$

Une autre façon consiste à prendre le vecteur $\overrightarrow{O_o O_n}$ définissant l'origine de l'effecteur et de lui ajouter, pour l'orientation, deux vecteurs quelconques de ceux du repère R_n (3) :

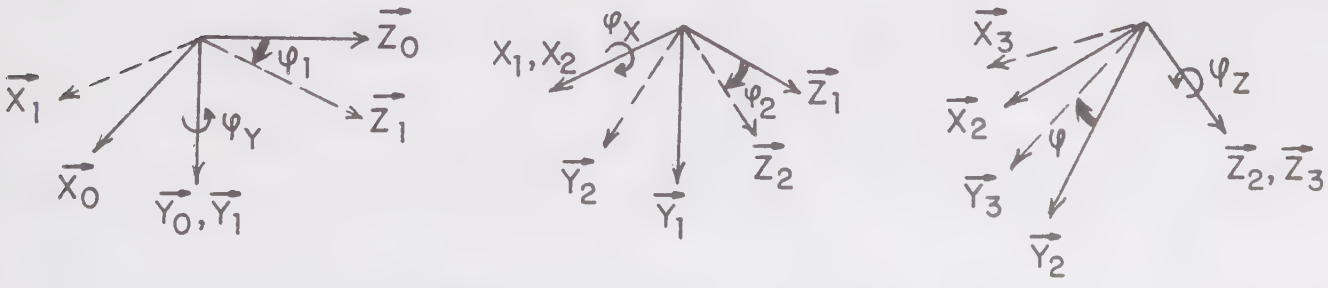
$$E = \{ \overrightarrow{O_o O_n}(\Theta), \vec{Y}(\Theta), \vec{Z}(\Theta) \}$$

D'autres représentations (11, 6) utilisent des paramètres dépendants comme les quaternions (12) ou les paramètres d'Olinde-Rodrigues (11).

Ces diverses représentations sont souvent adoptées à cause de la simplicité qu'elles introduisent dans la résolution du système global. Mais elles se heurtent à des difficultés quand il s'agit d'associer (pour la programmation des mouvements, par exemple) à une conjoncture entre l'effecteur et l'objet, les paramètres d'orientation nécessaires. Nous reprendrons dans la suite les variantes de représentations d'angles d'Euler (36, 19, 20) qui fournissent trois angles d'orientation indépendants et simples à interpréter (vu par le programmeur) pour caractériser l'orientation de l'effecteur.

I.3.2 - Représentation utilisant une variante des angles d'Euler

La partie terminale de la chaîne ou l'effecteur peut être caractérisé simplement par trois angles de rotation définissant complètement la matrice $M_n(\Theta)$. Le choix de ces angles doit être motivé par la possibilité de les déduire de façon simple à partir d'une conjoncture entre l'effecteur et l'objet à manipuler. Une façon simple pour préciser ce choix est de prendre les trois angles de rotation autour des trois axes de R_o (repère fixe).



où Φ_1 est l'angle $(\vec{Z}_o, \vec{Z}_1)$ lors d'une rotation autour de l'axe $\vec{Y}_1$

Φ_2 est l'angle $(\vec{Z}_1, \vec{Z}_2)$ lors d'une rotation autour de l'axe $\vec{X}_1$

Φ_3 est l'angle $(\vec{Y}_2, \vec{Y}_3)$ lors d'une rotation autour de l'axe $\vec{Z}_2$.

Soient M_X, M_Y et M_Z les matrices de rotation élémentaires définies par :

$$M_X = \begin{bmatrix} 1 & 0 & 0 \\ 0 & C\Phi_2 & -S\Phi_2 \\ 0 & S\Phi_2 & C\Phi_2 \end{bmatrix}, \quad M_Y = \begin{bmatrix} C\Phi_1 & 0 & S\Phi_1 \\ 0 & 1 & 0 \\ -S\Phi_1 & 0 & C\Phi_1 \end{bmatrix}, \quad M_Z = \begin{bmatrix} C\Phi_3 & -S\Phi_3 & 0 \\ S\Phi_3 & C\Phi_3 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

La matrice caractérisant l'orientation de l'effecteur est le produit des trois matrices élémentaires précédentes. Nous avons :

$$M_n(\Theta) = \phi(\Phi_1, \Phi_2, \Phi_3) = M_Y(\Phi_1)\ M_X(\Phi_2)\ M_Z(\Phi_3)$$

En développant le produit $M_Y.M_X.M_Z$ nous retrouvons la fonction matricielle $\phi(\Phi_1, \Phi_2, \Phi_3)$ de rotation globale :

$$\phi(\Phi_1, \Phi_2, \Phi_3) = \begin{bmatrix} C1C3 + S1S2S3 & -C1S3 + S1S2C3 & S1C2 \\ C2S3 & C2C3 & -S2 \\ -S1C3 + C1S2S3 & S1S3 + C1S2C3 & C1C2 \end{bmatrix}$$

Cette relation n'est pas bijective :

$$(\Phi_1, \Phi_2, \Phi_3) \underset{\text{infinité de solutions}}{\overset{\text{solution unique}}{\rightleftarrows}} \phi(\Phi_1, \Phi_2, \Phi_3)$$

En effet, cette correspondance admet des solutions de la forme :

$$\phi(\Phi_1,\Phi_2,\Phi_3) = \phi(\Phi_1 + K\Pi,\ \Pi - \Phi_2,\ \Phi_3 + K\Pi)$$

Lorsque $\Phi_i \in [-\Pi, +\Pi]$, i : 1, 2, 3, nous avons les deux solutions possibles :

$$\Phi = \{\Phi_1,\Phi_2,\Phi_3\} \quad \text{et} \quad \Phi^* = \{\Phi_1 - S(\Phi_1).\Pi,\ \Pi - \Phi_2,\ \Phi_3 - S(\Phi_3).\Pi\}$$

où S désigne la fonction signe.

L'inversion de ce système peut se faire en respectant un critère de choix. Un tel critère peut être obtenu en imposant la continuité de la trajectoire poursuivie par le robot. Soit $D(\Phi)$ une distance quelconque et désignons par Φ_I la solution précédente. La comparaison de $D\ (\Phi - \Phi_I)$ à $D\ (\Phi^* - \Phi_I)$ permet de sélectionner la solution qui assure la continuité de la trajectoire.

D'autre part pour $\Phi_2 = \pm\ \Pi/2$ la fonction $\phi(\Phi_1, \pm\ \Pi/2, \Phi_3)$ n'est fonction que de $\Phi_1 + \Phi_3$ ou de $\Phi_1 - \Phi_3$. Par conséquent la connaissance de ϕ ne suffit pas pour déterminer les angles Φ_1 et Φ_3. En opérant, avec le même critère de continuité, une solution pour Φ_1 (ou Φ_3) peut être obtenue par extrapolation polynomiale sur les états précédents (à l'ordre 3 ou 4). L'angle Φ_3 (ou Φ_1) sera ensuite déterminé par la connaissance de $\Phi_1 + \Phi_3$ et la solution obtenue par continuité pour l'un des deux.

I.3.2.1 - Définition du vecteur de l'effecteur

Le vecteur E caractérisant l'état de l'effecteur est défini par

$$E = (x,\ y,\ z,\ \Phi_1,\ \Phi_2,\ \Phi_3)^t$$

Pour toutes configurations de E existent un vecteur $\overrightarrow{O_o O_n} = (x,y,z)^t$ et une matrice $M_n(\Theta) = \phi(\Phi_1,\Phi_2,\Phi_3)$ déterminant la position et l'orientation de l'effecteur. A l'inverse (voir annexe 1) la connaissance de la matrice $M_n(\Theta)$ permet la déter-

mination, par continuité de la trajectoire, d'un triplet unique (Φ_1,Φ_2,Φ_3). Ce résultat est général car il s'applique à toute variante de représentation utilisant des combinaisons quelconques d'angles d'Euler.

I.4 - INVERSION DU MODELE GEOMETRIQUE

Généralement le robot dispose de mécanismes de régulation (en position ou en vitesse) au niveau de ces variables articulaires. Comme la tâche est décrite en termes d'évolution du vecteur de l'effecteur E il est donc possible, pour réaliser la commande, de chercher à associer à ce dernier un vecteur de configuration articulaire θ . Il s'agit de calculer la solution θ pour réaliser la transformation de coordonnées inverse $\theta = F^{-1}(E)$.

I.4.1 - Problématique du modèle géométrique inverse (annexe 1)

Nous avons vu dans le paragraphe précédent que la correspondance entre les angles d'orientation de l'effecteur et les vecteurs de R_n ne pose pas de difficultés majeures. Le problème se réduit à l'étude des équations non linéaires décrites dans le vecteur $\overrightarrow{O_o O_n}(\theta)$ et la matrice $M_n(\theta)$. Il s'agit de 9 équations non linéaires à n inconnues (θ_i, i : 1,..,n). Ces dernières sont composées de rotations (liaisons rotoïdes) ou de translations (liaisons prismatiques). Elles se présentent, dans le système non linéaire en cos (θ_i) et sin (θ_i) dans le premier cas et en θ_i dans le second cas.

La matrice $M_n(\theta)$ est engendrée à partir de trois variables indépendantes. La représentation $\{\overrightarrow{O_o O_n}(\theta), M_n(\theta)\}$ est engendrée à partir de 6 variables indépendantes (l'effecteur est un point orienté dans $\mathbb{R}^3$). Le système est redondant lorsque le nombre de ces variables articulaires est supérieur à 6. Dans ce cas on lui adjoint des contraintes (linéaires ou

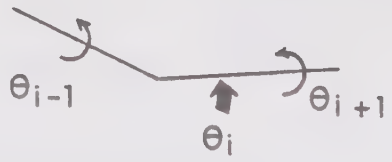
quadratiques) pour ramener le rang des variables articulaires à 6. Ces contraintes sont fonctionnelles et permettent de résoudre des problèmes intermédiaires pour éviter des obstacles (au niveau d'une articulation) ou pour remplacer les variables indépendantes qui arrivent en butée mécanique. De façon générale la chaîne articulée se ramène à 6 degrés de liberté indépendants.

La solubilité du système a été étudiée par PIEPER et ROTH (8, 9) qui ont montré la possibilité d'obtenir une solution lorsque les axes de deux liaisons successives sont soit parallèles soit concourantes et perpendiculaires. Une telle condition se révèle vérifiée pour la quasi totalité des robots actuels. Ces chercheurs ont calculé en outre les expressions des trois premières variables articulaires pour toutes les configurations de structures possibles.

I.4.2 - Unicité et singularité

L'unicité de la solution n'est pas assurée en général. Soit Θ_i une variable articulaire (rotoïde) ayant un domaine qui comprend une partie symétrique.

Lorsque les variables Θ_{i-1} et Θ_{i+1} sont des rotations ayant des domaines supérieurs à Π radians alors deux solutions sont possibles :



$$\{\Theta_{i-1},\ \Theta_i,\ \Theta_{i+1}\}$$

$$\{\Theta_{i-1} - S(\Theta_{i-1})\Pi,\ -\Theta_i,\ \Theta_{i+1} - S(\Theta_{i+1})\Pi\}$$

où S est la fonction signe.

La solution est alors obtenue sur la base d'un critère assurant la continuité de la trajectoire. L'examen de la distance qui sépare la solution initiale aux deux so-

lutions obtenues conduit à un choix sans ambiguité lorsque $\Theta_i \neq 0$.

Au cours du mouvement il peut arriver que deux axes de rotation soient alignés. Le système ne dépend plus explicitement de ces deux rotations mais de leur somme (par exemple, lorsque $\Theta_i = 0$). Une infinité de choix est alors possible pour déterminer ces deux rotations (Θ_{i-1} et Θ_{i+1}). En termes de rang les variables articulées (Θ_j, j : 1,...,n), par leur dépendance avec l'effecteur, sont de rang (n-1). Plusieurs moyens sont possibles pour déterminer explicitement ces deux rotations :

- choisir les solutions qui éloignent ces degrés de liberté des butées
- utiliser un critère de vraisemblance (quadratique) par rapport à l'état précédent
- effectuer une extrapolation polynomiale (par rapport à l'évolution précédente) pour l'une des variables indéterminées.

Dans ce dernier cas la solution obtenue n'introduit pratiquement pas de discontinuité sur la trajectoire si le degré de l'approximation est suffisamment grand (à l'ordre 4 par exemple).

I.4.3 - Remarques et conclusion

L'inversion du modèle géométrique nécessite la connaissance de toutes les coordonnées de la tâche même lorsque ces dernières ne sont pas toutes modifiées au cours d'un trajet particulier. Mais cela ne présente pas d'inconvénient dans la mesure où la prise en charge des paramètres fournis par l'utilisateur (leur nombre va de 1 à 6) conduit, par une procédure générale, à retrouver les grandeurs instantanées $\overrightarrow{O_o O_n}$ et M_n. Nous verrons plus loin, dans le cadre de notre application, que les fonctions de mouvement sont engendrées par un module unique qui permet de retrouver toutes les données nécessaires

à l'inversion quel que soit le mouvement concerné.

Cette représentation géométrique ne pose pas de problème à l'utilisation d'angles indépendants pour l'orientation de l'effecteur ce qui constitue un point intéressant, car simple. L'emploi d'un modèle variationnel bien que plus intéressant conduit souvent à des représentations, de l'orientation, redondantes et difficiles à manipuler par l'utilisateur.

Bien que la solubilité puisse être vérifiée pour un robot, l'obtention de son modèle géométrique inverse n'est pas facile dans le cas général.

I.5 - MODELE VARIATIONNEL ET PROBLEME DE L'INVERSE

I.5.1 - Introduction

Nous avons vu dans la section précédente l'établissement d'une loi de coordination à partir du modèle géométrique. Les termes de cette loi ainsi que sa commodité ont été traités et discutés en fonction de la solubilité du système d'équations non-linéaires qu'est le modèle géométrique d'une chaîne articulée. Bien que pour une majorité de robots (90 %) une inversion du modèle géométrique soit possible, un intérêt particulier s'est développé autour de la représentation variationnelle ou cinématique.

D.E. WHITNEY, en 1969, a proposé une loi de commande relative aux vecteurs de vitesse, de translation, et de rotation du segment terminal d'un robot. Cette représentation conduit à une linéarisation de la loi de commande. Le problème de la coordination sans contraintes est celui de la résolution d'un système d'équations linéaires liant les paramètres de représentation de la tâche aux variables généralisées. Nous développons dans ce paragraphe un formalisme permettant l'obtention du modèle cinématique et variationnel. Le problème de l'inversion de ces modèles est ensuite discuté.

I.5.2 - Modèle cinématique et variationnel

Soit $\Omega_n(R_o ; R_o)$ le vecteur vitesse de rotation, du nième segment, par rapport à R_o et exprimé dans ce dernier référentiel, nous avons :

$$\Omega_n(R_o ; R_o) = \sum_{i=1}^{n} \omega_i(R_i ; R_o)$$

où $\omega_i(R_i ; R_o)$ est le vecteur de vitesse de rotation du à θ_i (si elle est rotoïde) par rapport à R_i et exprimé dans le référentiel R_o. En utilisant les matrices de changement de repère M_i nous avons :

$$\Omega_n(R_o ; R_o) = \sum_{i=1}^{n} M_o^i \, \omega_i(R_i ; R_i)$$

où $M_o^i = M_1, M_2, \ldots, M_i$ l'indice i est relatif à la variable généralisée θ_i. Nous désignons par M_i la matrice de rotation élémentaire quand θ_i est rotoïde ou la matrice identité quand θ_i est prismatique.

Le vecteur $\omega_i(R_i ; R_i)$ est le vecteur vitesse de rotation engendré par la variable généralisée rotoïde θ_i. Par exemple si l'axe $\vec{OZ}$ de R_i coïncide avec l'axe du ième segment alors ω_i s'écrit :

$$\omega_i(R_i ; R_i) = (0, 0, \dot{\theta}_i)^t$$

Soit $\dot{\theta}$ le vecteur de la dérivée par rapport au temps du vecteur des variables généralisées θ. Le vecteur Ω_n s'écrit :

$$\underset{(3,1)}{\Omega_n(R_o ; R_o)} = \underset{(3,n)}{J_\omega(\theta)} \cdot \underset{(n,1)}{\dot{\theta}}$$

La matrice $J_\omega(\theta)$ est engendrée par les cos (θ_i) et sin (θ_i) contenus dans les matrices M_i (i : 1,..., n).

Désignons par $V(O_n/R_o ; R_o)$ le vecteur vitesse de

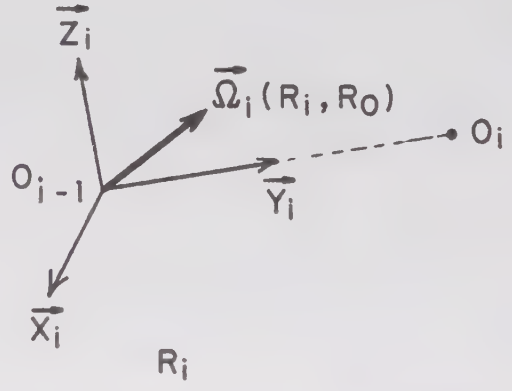
translation du point O_n, extrémité du segment terminal, calculé par rapport à R_o et exprimé dans ce dernier repère, nous avons

$$V(O_n/R_o\ ;\ R_o) = \frac{d}{dt}\{\overrightarrow{O_oO_n}\}_{/R_o} = \sum_{i=1}^{n} \frac{d}{dt}\{\overrightarrow{O_{i-1}O_i}\}_{/R_o}$$

Le vecteur V est donc la résultante des vecteurs vitesse dues aux translations des sommets de chaque segment de la chaîne. Le vecteur $\frac{d}{dt}\{\overrightarrow{O_{i-1}O_i}\}_{/R_o}$ s'exprime de façon simple dans son propre référentiel R_i

$$\frac{d}{dt}\{\overrightarrow{O_{i-1}O_i}\}_{/R_o} = M_o^i \frac{d}{dt}\{O_{i-1}O_i\}_{/R_i}$$

Exprimé par rapport à R_i ce vecteur est la somme de deux vecteurs indépendants : le premier est dû à la présence en O_{i-1} du vecteur de vitesse de rotation $\Omega_i(R_i\ ;\ R_o)$ et le second est dû à la translation L_i de O_i par rapport à O_{i-1} (lorsque Θ_i est prismatique) :



$$\frac{d}{dt}\{O_{i-1}O_i\}_{/R_i} = \Omega_i(R_i\ ;\ R_o) \wedge L_{i/R_i} + \overset{\circ}{L}_{i/R_i}$$

où "$\wedge$" désigne le produit vectoriel

$\overset{\circ}{L}_{i/R_i}$ est engendré par le degré de liberté prismatique Θ_i. Par exemple dans le cas de la figure nous avons $\overset{\circ}{L}_i = (0, \overset{\circ}{\Theta}_i, 0)^t$.

Nous obtenons l'expression de $V(O_n/R_o\ ;\ R_o)$:

$$V(O_n/R_o\ ;\ R_o) = \sum_{i=1}^{n} M_o^i \{\Omega_i(R_i\ ;\ R_o) \wedge L_{i/R_i} + \overset{\circ}{L}_{i/R_i}\}$$

Compte tenu du fait que Ω_i et $\dot{L}_i$ s'expriment en fonction des dérivées des variables généralisées le vecteur $V(O_n/R_o ; R_o) = V_n$ s'écrit :

$$V_n = J_V(\Theta) \cdot \dot{\Theta}$$

$$(3,1)\ (3,n)\quad (n,1)$$

En groupant les vecteurs Ω_n et V_n nous avons la forme condensée suivante :

$$\begin{bmatrix} \Omega_n \\ V_n \end{bmatrix} = \begin{bmatrix} J_\omega(\Theta) \\ \hline J_v(\Theta) \end{bmatrix} \cdot \dot{\Theta} \qquad (I\text{-}1)$$

$$(6,1)\quad (6,n)\quad (n,1)$$

Soit $\dot{E}$ le vecteur dérivé par rapport au temps du vecteur de l'effecteur. Ce dernier est lié au vecteur $[\Omega_n, V_n]$ par une matrice de changement de repère P(E) :

$$\dot{E} = P \cdot \begin{bmatrix} \Omega_n \\ V_n \end{bmatrix}$$

Le modèle cinématique se déduit de ces deux dernières relations :

$$\dot{E} = P \begin{bmatrix} J_\omega \\ \hline J_v \end{bmatrix} \cdot \dot{\Theta} = J \cdot \dot{\Theta}$$

$$(m,1) \qquad (m,n)(n,1)$$

Le modèle variationnel s'obtient à partir du modèle cinématique en assimilant, lorsque ΔE est faible, le vecteur $\dot{E}$ au vecteur $\frac{1}{\Delta T}\ \Delta E$ (ΔT est la période de la commande), nous avons :

$$\Delta E = J \cdot \Delta\Theta$$

$$(m,1)\quad (m,n)(n,1)$$

I.5.3 - Inversion du modèle variationnel

L'élaboration d'une loi de coordination par le modèle variationnel revient à inverser le système linéarisé représenté par l'équation $\Delta E = J\Delta\Theta$. La solution, quand elle existe, permet de convertir un écart au niveau des paramètres de la tâche en un autre au niveau des variables généralisées :

$$\underset{(n,1)}{\Delta\Theta} = \underset{(n,m)}{J^{-1}(\Theta)} \cdot \underset{(m,1)}{\Delta E}$$

Les propriétés des systèmes linéaires s'appliquent au système d'équations $\Delta E = J.\Delta\Theta$. Ce dernier admet une solution unique $\Delta\Theta$ lorsque le nombre de degrés de liberté du manipulateur est égal à celui des paramètres de la tâche (n=m=6) et quand ces deux groupes de variables sont séparément indépendants. Cependant des difficultés sont manifestes dès que le système se trouve, au moins, dans l'une des situations suivantes :

- Le manipulateur est redondant vis-à-vis de la tâche ; les m paramètres de la tâche ne sont pas indépendants.

- La tâche est redondante vis-à-vis du manipulateur ; au moins deux variables généralisées sont dépendantes par rapport à la représentation de l'effecteur.

Plus généralement lorsque le degré de liberté du manipulateur est plus grand que celui de la représentation (un point orienté dans $\mathbb{R}^3$ requiert 6 degrés de liberté) le système $\Delta E = J\Delta\Theta$ admet une infinité de solutions $\Delta\Theta$. Dans ces conditions, le rang r de la matrice J est au plus égal à m.

Plusieurs méthodes d'inversion ont été proposées (13).

La méthode des variables principales (3) conduit à extraire de la matrice $J_{m,n}$ le mineur de rang r le plus grand. Cela revient à limiter le mouvement (ou la mobilité du mani-

pulateur) sur r variables généralisées. Cette méthode s'appuie sur un choix de r degrés de liberté qui ne sont pas toujours les mêmes notamment dans le cas d'une singularité ou de l'arrivée en butée mécanique sur l'une des variables généralisées...

Une autre méthode consiste à minimiser un critère quadratique sur l'énergie (14, 15, 16) cinétique et potentielle. Le critère considéré s'écrit pour le cas de l'énergie cinétique :

$$Q = E^t.A.E$$

où A est une matrice définie positive donc régulière (la matrice d'inertie). On adjoint au système $\Delta E = J\Delta\Theta$ une équation homogène $C.\Delta E = 0$ de rang P qui représente le gradient de la fonction Q. La solution $\Delta\Theta$ qui minimise le critère Q peut être obtenue lorsque les p contraintes extérieures sont linéairement indépendantes et la matrice J est de rang n (en dehors des singularités du manipulateur), dans ce cas cette dernière s'écrit :

$$\Delta\Theta = DJ^t (JDJ^t)^{-1} \Delta E$$

où D est une matrice de rang n si les p contraintes, liées au critère, sont indépendantes.

Le critère de l'énergie cinétique est de nature dynamique et est appliqué à une commande cinématique pour laquelle il est difficilement exploitable. Il conduit en outre à l'inversion de deux matrices de rang n.

En utilisant les propriétés des matrices inverses généralisées A. FOURNIER (3,17) a proposé une méthode intéressante pour l'obtention d'une solution générale au système $\Delta E = J\Delta\Theta$. Cette méthode s'appuie dans sa description sur des notions d'algèbre linéaire. L'opérateur linéaire $J_{n,m}$ transforme un vecteur $\Delta\Theta \in \mathbb{R}^n$ en un vecteur $\Delta X \in \mathbb{R}^m$, soit $J^+_{m,n}$ une inverse généralisée réflexive de J. Le vecteur $\Delta\Theta^+ = J^+\Delta E$ est une projection sur l'espace image J_m (J^+) parallèlement au noyau N(J).

Le vecteur $\Delta\theta^- = (J^+J - I_n)Z$ est une projection du vecteur $Z \in \mathbb{R}^n$ sur le noyau N(J) parallèlement à l'espace image J_m (J^+J). D'autre part l'inverse généralisée J^+ de J permet d'écrire :

$$\text{rang } \{J, \Delta E\} = \text{rang } \{J^+\} = \text{rang } \{J\} = \text{rang } \{J^+J\} = r$$

Par conséquent il intervient dans la solution (n-r) équations linéairement indépendantes(rang $(J^+J - I_n) = n - r$). Le vecteur $\Delta\theta^+ = J^+ \Delta E$ est une solution qui appartient à un sous espace complémentaire de N(J) dans $\mathbb{R}^n$, il est donc linéairement indépendant de $\Delta\theta^- = (J^+J - I_n)Z$ quel que soit $Z \in \mathbb{R}^n$.

La solution générale du système $\Delta E = J\Delta\theta$ est de la forme :

$$\Delta\theta = J^+ \Delta E + (J^+J - I_n)Z$$

L'espace des solutions est de dimension n-r+1 pour tout ΔE non nul (et de dimension n-r lorsque $\Delta X = 0$). Le vecteur Z est le gradient d'une fonction scalaire définie positive sur $\Delta\theta$. Compte tenu de l'indépendance des solutions $\Delta\theta^+$ et $\Delta\theta^-$ l'optimisation de cette fonction critère est indépendante de l'état de la tâche ΔE.

En résumé, cette méthode propose une solution composée de deux parties découplées : la première est dépendante de la tâche ($\Delta\theta^+$) et la seconde est dépendante d'un critère d'optimisation. Son intérêt est d'avoir découplé, dans la solution, toute contrainte d'optimisation de l'état de la tâche.

I.5.4 - Conclusion

Les critères d'optimisation jouent un rôle important dans les systèmes de coordination. Ils sont utilisés pour aplanir les problèmes suivants :

- la redondance du manipulateur et de la représentation de l'effecteur

- les problèmes aux limites (les butées mécaniques)

- le problème des singularités de dépendance des variables généralisées et de la représentation de l'effecteur.

Ces considérations introduisent un couplage, entre les équations traduisant l'optimisation et celles traduisant la tâche, complexe dans le système de coordination. Il en découle que la continuité et la précision des trajectoires restent des éléments qui dépendent des optimisations faites pour chacun des problèmes cités plus haut. Bien que pour la plupart des robots actuels, de par leur architecture, ces problèmes peuvent être surmontés avec les méthodes variationnelles classiques, la méthode de l'inverse généralisée (3, 17) semble être la mieux adaptée au cas général à cause du découplage qu'elle présente, dans la solution, entre la tâche et l'optimisation.

I.6 - SYNTHESE D'UN SYSTEME DE COORDINATION GENERAL

Nous avons vu dans les trois premières sections les éléments fondamentaux de la fonction de coordination. Les modèles géométrique et variationnel définissent les relations entre les variables généralisées et les données de base du segment terminal du robot. La représentation de l'effecteur offre un moyen simple et optimisé pour définir l'état de ce dernier. La tâche est décrite aux termes de l'évolution dans le temps du vecteur de l'effecteur vis-à-vis d'un objectif à atteindre (transport d'un objet, suivi de trajectoires) ou vis-à-vis de la réaction du milieu à l'état précédent (mécanismes rebouclés).

L'analyse des interactions entre les différents modes de commande nous conduit, dans ce paragraphe, à définir une structure générale orientée vers la programmation automatique. Nous traiterons, à ce sujet, les problèmes liés aux éléments de base permettant d'engendrer les mouvements d'une façon simple et structurée. Une application de ce système de coordination est présentée dans le cadre d'une implantation sur microprocesseurs pour la commande d'un manipulateur à six degrés de liberté.

I.6.1 - Définition d'un système de coordination des mouvements

La tâche demandée au robot est souvent constituée d'une phase de transport de l'objet qui est suivie d'une autre phase de manipulation de ce dernier. Si la première phase peut être spécifiée par l'état de l'effecteur E/R_o la seconde ne l'est pas. En effet l'opération de manipulation de l'objet, se fait directement par rapport à une partie de l'objet soit parce que cette dernière entre en contact avec un autre objet, soit parce que le centre d'action voulu par l'utilisateur est lié à cette partie. Au niveau du système de coordination cela revient à distinguer les actions directes sur le vecteur de l'effecteur (première phase) des actions très variables qui agissent sur un référentiel défini, par l'utilisateur, à un endroit particulier de l'objet. Cela revient à définir un segment supplémentaire (l'objet) attaché rigidement au segment terminal.

Ces remarques nous ont conduit à définir une structure (fig. I.1) de coordination assurant la compatibilité des différentes actions possibles. Cette structure comprend principalement trois modules de base :

- Le module de changement de coordonnées engendré par la représentation géométrique ou variationnelle.

- Le module de liaison de la représentation de l'effecteur et les données de base du segment terminal. Ce dernier a la charge d'engendrer les trajectoires, selon une loi donnée, faisant passer le système de l'état initial E_o à l'état désiré E_d.

- Le module de liaison et de définition du point d'action lié à l'objet. Il assure,en outre, la compatibilité du système en calculant l'état équivalent de l'effecteur.

Nous expliciterons les fonctions principales de ces deux derniers modules :

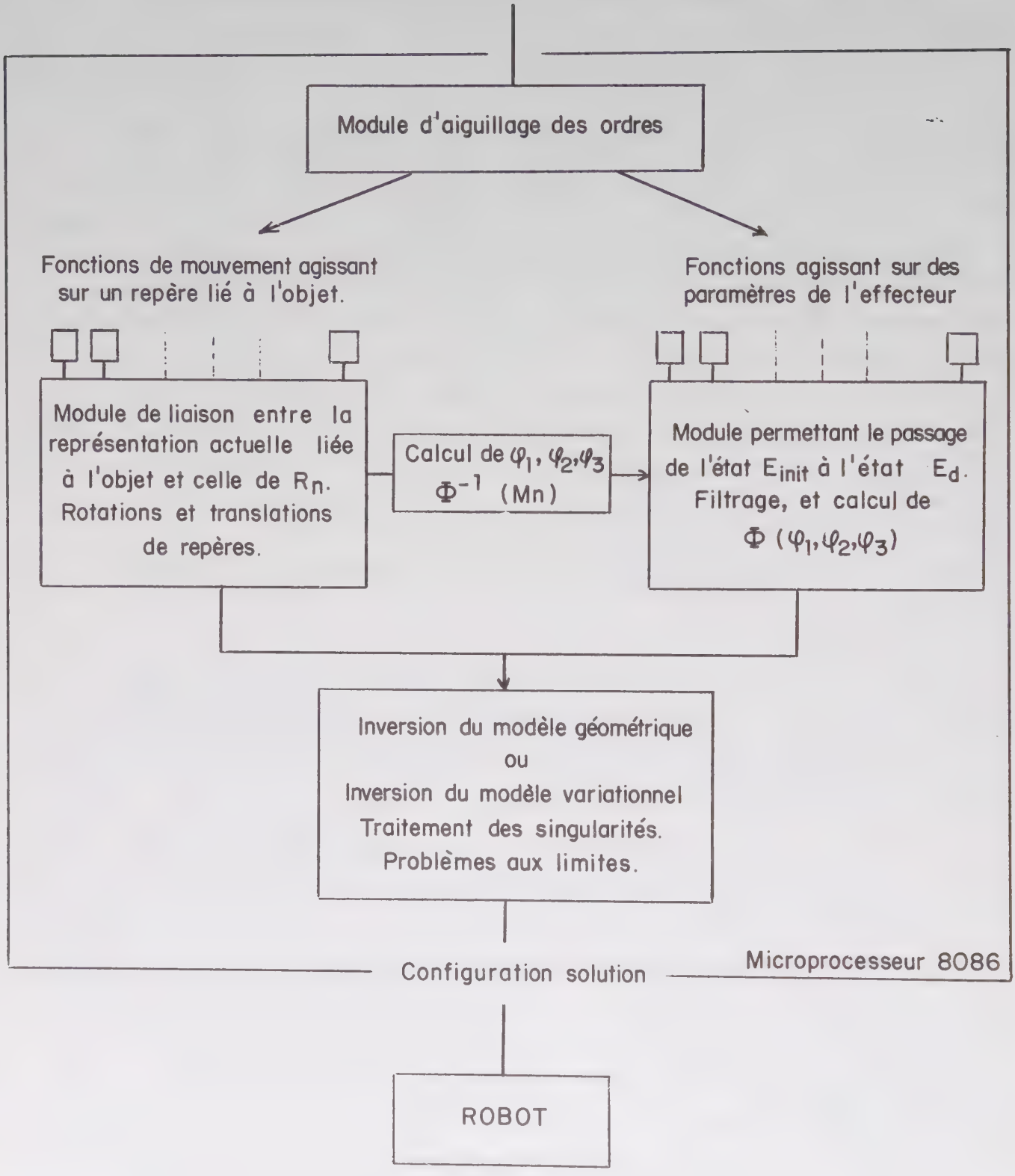


Figure I_1 : Schéma du système de coordination des mouvements.

Passage de l'état initial E_o à l'état désiré E_d

Ces fonctions sont engendrées par une forme générale de mouvement permettant le passage de l'effecteur de l'état initial E_o à l'état désiré E_d. Il s'agit de déterminer une loi discrète d'évolution vérifiant les conditions suivantes :

$$E(0) = E_o \; ; \; E(NT) = E_d \; ; \; \dot{E}(0) = \dot{E}(NT) = \ddot{E}(0) = \ddot{E}(NT) = 0$$

où $\dot{E}$ et $\ddot{E}$ désignent respectivement les dérivées premières et secondes discrètes

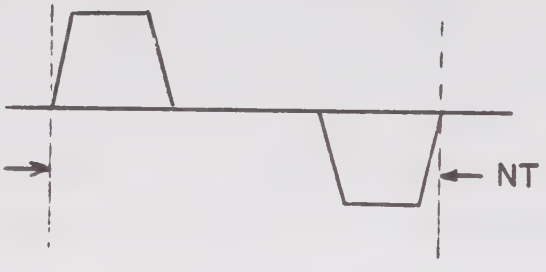
T est la période élémentaire de la commande

NT le temps mis pour effectuer le mouvement.

Soit R(KT) un filtre discret obtenu à partir d'une loi imposée sur la dérivée seconde discrète (voir figure) de sa sortie. Il vérifie en outre les tolérances de vitesse et d'accélération imposées par la mécanique. Sa grandeur de sortie satisfait les normes imposées plus haut au vecteur E et converge vers l'unité.

Pour une coordination utilisant le modèle géométrique F nous avons à l'instant KT :

$$E(KT) = E_o + R(KT).(E_d - E_o)$$



Evolution de la dérivée seconde discrète

L'évolution du vecteur des variables généralisée est obtenue par le modèle géométrique :

$$\Theta(KT) = F^{-1} \; (E \; (KT) \;)$$

Pour une coordination avec le modèle variationnel nous avons :

$$\Delta E(KT) = \underbrace{E_o + R(KT).(E_d - E_o)}_{\text{état désiré}} - F \; (\Theta_r(KT) \;)$$

où $\Theta_r(KT)$ est le vecteur des variables généralisées relevé sur le robot à l'instant KT (K : 1,...,N).

L'évolution du vecteur des variables généralisées est donnée par l'inverse du modèle variationnel :

$$\Theta(KT) = \Theta_r(KT) + J^{-1} \ (\Delta E \ (KT) \ )$$

Fonctions de liaison avec un repère extérieur

1) Le référentiel d'action se déduit de R_n par une translation. Pour agir par rapport à un endroit précis de l'objet, tenu par l'effecteur, le référentiel d'action R_A,est déterminé par R_n qui est translaté de $\vec{T}/R_n$ (par déclaration). Pour effectuer une rotation générale $\phi(\Phi_X,\Phi_Y,\Phi_Z)$ (définie selon I.2.2) autour des axes du repère R_A. Le vecteur $\overrightarrow{O_oO_n}$ et la matrice M_n subissent les transformations suivantes :

$$\overrightarrow{O_oO_n}/R_o \quad \longleftarrow \quad \overrightarrow{O_oO_n}/R_o + M_n \ (I_3 - \phi(\Phi_X,\Phi_Y,\Phi_Z)^T/R_n$$

devient

où I_3 est la matrice identique d'ordre 3.

Nous déduisons la configuration de l'orientation de l'effecteur par la relation suivante (I.2.2) :

$$\phi^{-1} \ (M_n) \rightarrow \Phi_1, \ \Phi_2, \ \Phi_3$$

2) Le référentiel d'action se déduit de R_n par une rotation.

Soit Φ_{XA}, Φ_{YA} et Φ_{ZA} les angles (I.2.2) de référence définissant le repère R_A par rapport à R_n. La translation de R_A selon le vecteur de commande $\vec{L}/R_A$ conduit au calcul du vecteur $\overrightarrow{O_oO_n}$ et la matrice M_n par les relations suivantes :

$$\overrightarrow{O_oO_n}/R_o \quad \longleftarrow \quad \overrightarrow{O_oO_n} + M_n . \ \phi(\Phi_{XA}, \ \Phi_{YA}, \ \Phi_{ZA}) . \vec{L}/R_A$$

3) Le référentiel d'action se déduit de R_n par une rotation et une translation. Pour un mouvement de translation $\vec{L}/R_A$ nous avons :

$$\overrightarrow{O_o O_n} \longleftarrow \overrightarrow{O_o O_n} + M_n \, (\phi_A \cdot \vec{L}/R_A + \vec{T}/R_n)$$

Le référentiel R_n demeure inchangé. Pour une rotation générale $\phi(\Phi_{XA}, \Phi_{YA}, \Phi_{ZA})$ par rapport à R_A nous avons :

$$\overrightarrow{O_o O_n} \longleftarrow \overrightarrow{O_o O_n} + M_n \, \phi_A \, \{I_3 - \phi(\Phi_{XA}, \Phi_{YA}, \Phi_{ZA})\}\vec{T}/R_n$$

et

$$M_n \longleftarrow M_n \, (\phi_A \cdot \phi_{XA,YA,ZA} \cdot \phi_A^{-1})$$

Une table (fig. I-2) explicite les fonctions de coordination engendrées avec notre structure de commande.

Figure 1-2 : Table des fonctions de coordination en position

Nom de la fonction	Arguments	Description de l'action
DEPLG	$X, Y, Z, \Phi_1, \Phi_2, \Phi_3$	Mouvement vers l'état général E_d de l'effecteur
DEPL	X, Y, Z	Translation du réf. de l'effecteur/orientation fixe
DROTG	Φ_1, Φ_2, Φ_3	Rotation " " " / origine fixe
DMPX	ΔX	Mouvement de ΔX selon l'axe $\vec{OX}$
DMPY	ΔY	" " ΔY " " $\vec{OY}$
DMPZ	ΔZ	" " ΔZ " " $\vec{OZ}$
DROT1	$\Delta\Phi_1$	Rotation de $\Delta\Phi_1$ autour de l'axe $\vec{OY}$
DROT2	$\Delta\Phi_2$	" " $\Delta\Phi_2$ " " " $\vec{OX}$
DROT3	$\Delta\Phi_3$	" " $\Delta\Phi_3$ " " " $\vec{OZ}$
MEPX (2)	δX	Mouvement élémentaire de δX selon l'axe $\vec{OX}$
MEPY (2)	δY	" " " δY " " $\vec{OY}$
MEPZ (2)	δZ	" " " δZ " " $\vec{OZ}$
REPX (1)	$\delta\Phi_1$	Rotation élémentaire de $\delta\Phi_1$ autour de $\vec{OX}$
REPY (1)	$\delta\Phi_2$	" " " $\delta\Phi_2$ " " $\vec{OY}$
REPZ (1)	$\delta\Phi_3$	" " " $\delta\Phi_3$ " " $\vec{OZ}$
TRANSLAT	T_X, T_Y, T_Z	Déclaration de translation du réf. de l'effecteur (avec (1))
REPXTZ	$\delta\Phi_1, T_Z$	Rotation élémentaire de $\delta\Phi_1/\vec{OX}$, réf. translaté de $T_Z/\vec{OZ}$
REPYTZ	$\delta\Phi_2, T_Z$	" " $\delta\Phi_2/\vec{OY}$, réf. translaté de $T_Z/\vec{OZ}$
REPXTY	$\delta\Phi_1, T_Y$	" " $\delta\Phi_1/\vec{OX}$, réf. translaté de $T_Y/\vec{OY}$
REPZTY	$\delta\Phi_3, T_Y$	" " $\delta\Phi_3/\vec{OZ}$, réf. translaté de $T_Y/\vec{OY}$
ROTATION	Φ_X, Φ_Y, Φ_Z	Déclaration de rotation du réf. de l'effecteur (avec (2))
MEPXRZ	δ_X, Φ_Z	Translation élémentaire de $\delta X/\vec{OX}$, réf. tourné $\Phi_Z/\vec{OZ}$
MEPYRZ	δ_Y, Φ_Z	" " $\delta_Y/\vec{OY}$, réf. tourné $\Phi_Z/\vec{OZ}$
MEPXRY	δ_X, Φ_Y	" " $\delta_X/\vec{OX}$, réf. tourné $\Phi_Y/\vec{OY}$
MEPZRY	δ_Z, Φ_Y	" " $\delta_Z/\vec{OZ}$, réf. tourné $\Phi_Y/\vec{OY}$
MEPXYZ(1)	$\delta_X, \delta_Y, \delta_Z$	Mouvement élément. de $\delta_X, \delta_Y, \delta_Z$ selon $\vec{OX}$, $\vec{OY}$, $\vec{OZ}$
REPXYZ(2)	$\delta\Phi_1, \delta\Phi_2, \delta\Phi_3$	Rotation élémentaire de $\delta\Phi_1, \delta\Phi_2, \delta\Phi_3$ autour de $\vec{OX}$, $\vec{OY}$, $\vec{OZ}$

I.6.2 - Principaux modes de commande

Selon la méthode employée (Modèle géométrique ou variationnel) plusieurs normes de commande sont possibles (14, 18). Nous donnons le principe de base de telles lois de commande :

- Associer au vecteur de l'effecteur $E_d(+)$ le vecteur de configuration correspondant (8) par la relation $\theta(t) = F^{-1}(E_d(t))$. Ce dernier est alors considéré comme un vecteur de consigne de position pour une régulation au niveau des actionneurs. Par conséquent le vecteur de l'effecteur $E_r(t)$ du robot suit assez fidèlement (dépend de la régulation) le vecteur désiré $E_d(t)$ (voir le schéma de cette régulation Fig. I-3). On note que cette forme est caractérisée par une régulation en position au niveau des actionneurs.

- Associer à un vecteur d'écart ΔE, entre le vecteur désiré $E_d(t)$ et celui du robot $E_r(t) = F(\theta_r(t))$, un vecteur de consigne de vitesse $\dot{\theta}(t)$ au niveau articulaire :

$$\dot{\theta}(t) = \frac{1}{T} J^{-1} (E_d - E_r) \text{ quand } E_d \text{ est voisin de } E_r$$

où T est la période de commande

Le vecteur de consigne $\dot{\theta}(t)$ est imposé à l'entrée d'une régulation de vitesse au niveau des actionneurs (schéma fig. I-4).

- Associer au vecteur de l'écart $\Delta E(t)$ un vecteur de force corrective $\mathcal{F}(t)$ par la relation :

$$\mathcal{F}(t) = \Delta E(t) + \Delta\dot{E}(t) \text{ (proportionnel et dérivée)}$$

Le vecteur $\mathcal{F}(t)$ est converti ensuite en vecteur des couples moteurs-actionneurs $\Gamma_\theta(t)$ par la relation :

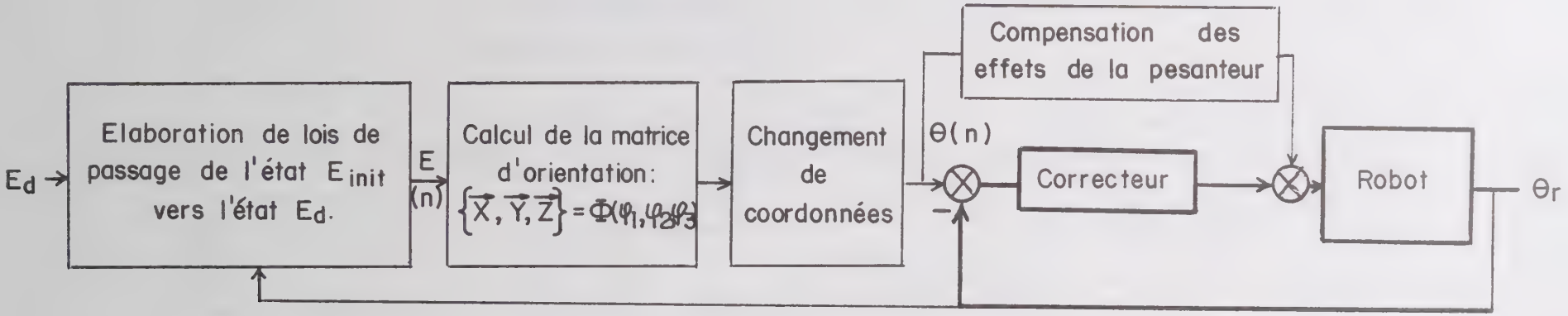
$$\Gamma_\theta(t) = J^t . \mathcal{F}(t)$$

On note que cette régulation est réalisée au niveau de l'effecteur (schéma Fig. I-5).

Pour résumer ces trois modes de commande l'évolution des grandeurs calculées en ligne s'établit de la façon suivante :

Modèle utilisé	Grandeurs calculées dans un cycle de commande
Géométrique	$\Theta(t) = F^{-1}(E(t))$
Variationnel	$E_r(t) = F(\Theta_r(t))$; $\dot{\Theta} = J^{-1}\,\Delta E$
"	$E_r(t) = F(\Theta_r(t))$; $\Gamma_\Theta = J^t . \mathcal{F}(t)$

Les effets de la gravité sont compensés, sur chaque articulation, dans le but de supprimer les erreurs statiques (voir les expressions des couples dus à la pesanteur en annexe 4).



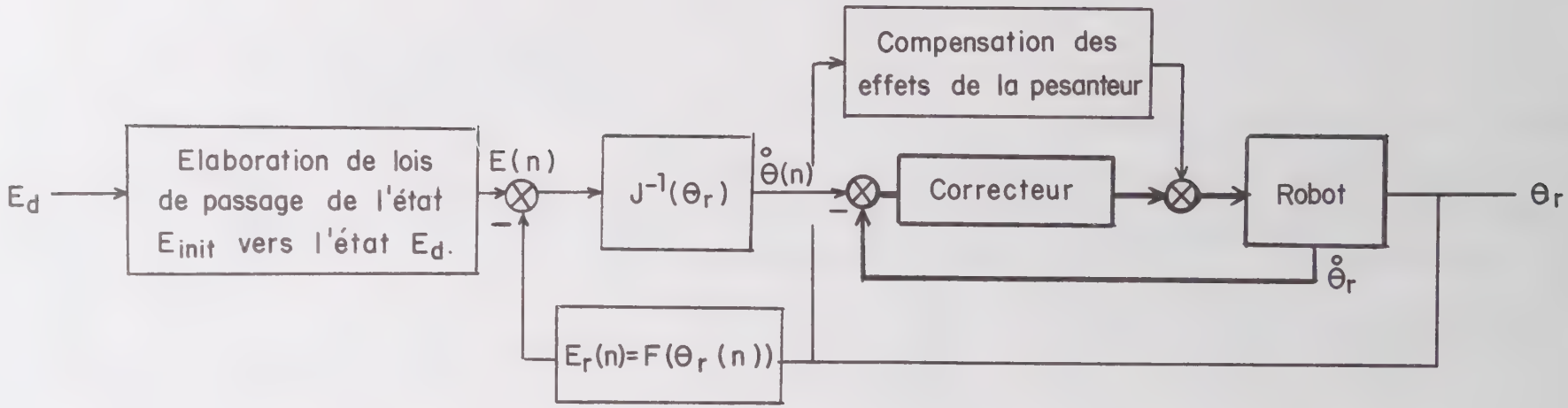
$E_d = (X, Y, Z, \varphi_1, \varphi_2, \varphi_3)^t$

$E(n) = E(nT)$; T période de commande

———— partie numérique.

▬▬▬▬ partie analogique et mécanique.

Figure I.3 _ Coordination de position avec le modèle geométrique.

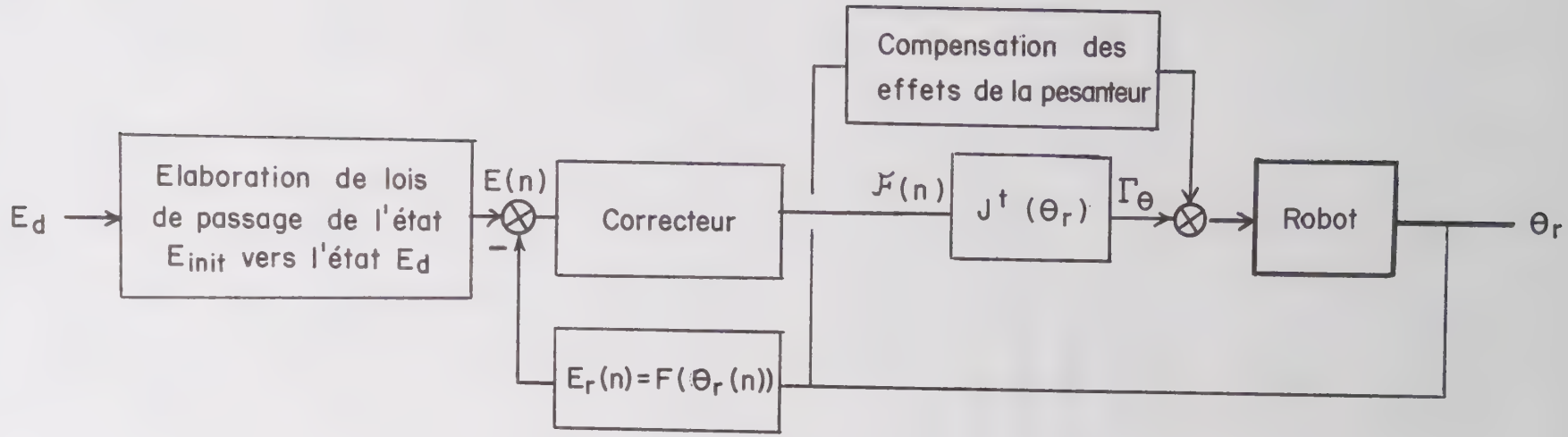


$\dot{\Theta}(n) = \frac{\Delta\Theta(nT)}{T}$; T période de commande.

_____ partie numérique.

_____ partie analogique et mécanique.

Figure I. 4 : Coordination de position avec le modèle variationnel.



$\mathcal{F}$ est le vecteur de force de correction.

———— partie numérique

———— partie analogique et mécanique.

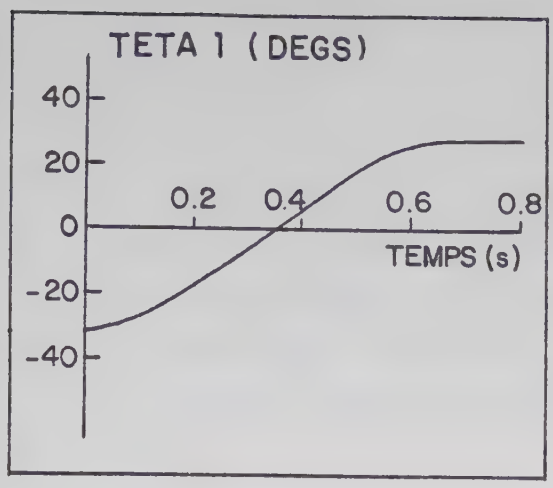
Figure I.5 _ Coordination de position utilisant la notion de force.

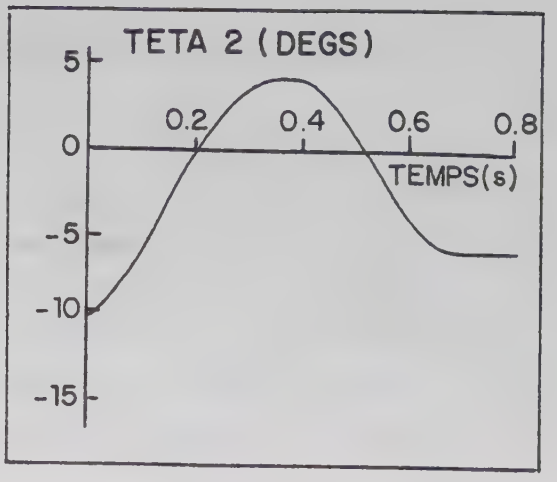
I.6.3 - Application à la coordination dans le cas d'un robot manipulateur

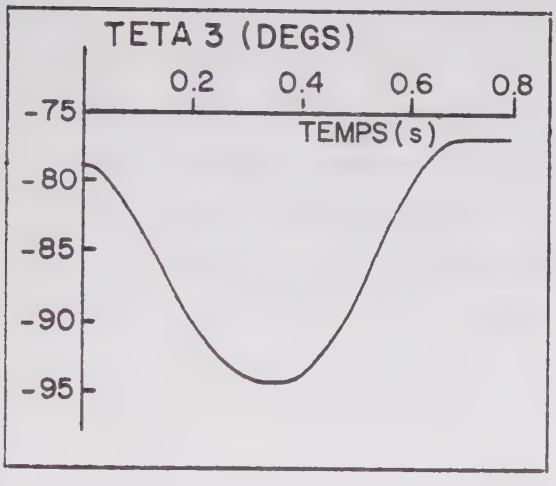
En application nous avons étudié notre système de coordination pour le cas du manipulateur M.A 23 (conçu au C.E.A. par le Dpt STEP/EMH). Ce dernier est muni de six degrés de liberté. Cette coordination fut testée et validée pour deux manipulateurs de même type qui diffèrent par leurs charges maximales (10 kg et 25 kg).

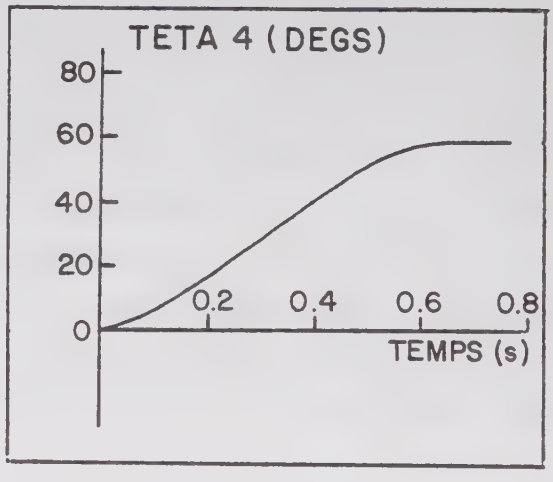
Sur le plan de l'implantation et de la commande en temps réel nous avons construit notre système sur une unité microprocesseur parmi les plus performantes sur la scène actuelle (INTEL 8086 avec son coprocesseur flottant 8087). L'optimisation des équations mathématiques et la constitution d'une bibliothèque de fonctions spécialisées et rapides nous ont permis d'obtenir une période de commande performante (8 ms) par rapport à celle obtenue sur les mini-calculateurs (entre 30 et 40 ms).

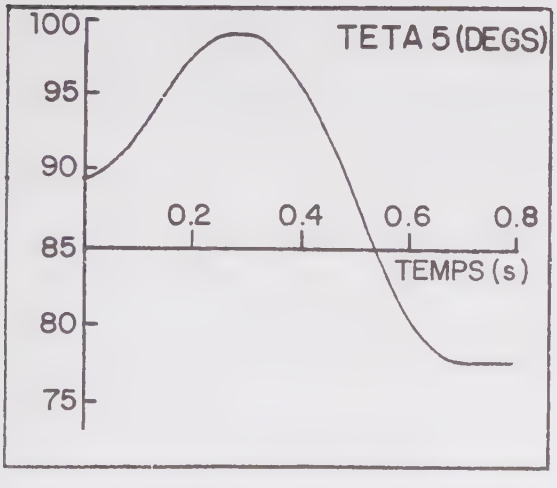
Les courbes présentées sur la figure (I-6) montrent la réponse du système de coordination (évolution des variables généralisées) vis-à-vis d'une commande portant sur un état désiré de l'effecteur.











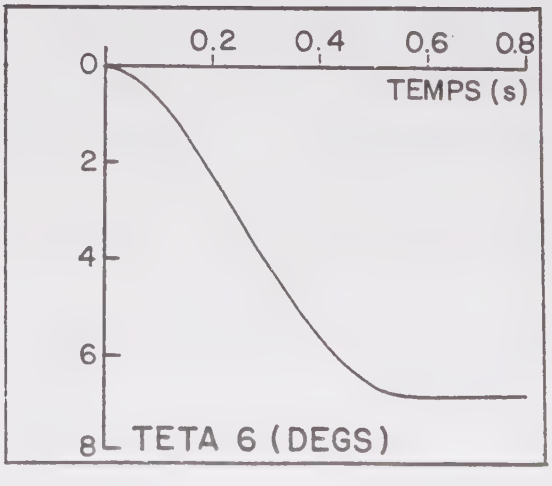


Figure I. 6 _ Evolution des angles géométriques.

I.7 - CONCLUSION SUR LA PREMIERE PARTIE

Dans le but de synthétiser des fonctions de coordination générales nous avons exposé dans cette partie les modèles géométrique et variationnel pour une structure articulée générale. Nous avons discuté ensuite les problèmes liés à l'établissement des modèles inverses. L'accent a été mis sur les heuristiques qui peuvent être employées pour rendre ces relations quasiment biunivoques.

Nous avons porté un intérêt particulier au problème de la structuration des fonctions de coordination. Ainsi nous avons proposé un système modulaire pour concevoir une telle coordination enrichie par de nombreuses fonctions de mouvements. Ces dernières présentent une marge d'éléments très utiles pour la programmation des tâches. Leur emploi permet d'agir, de façon simple, sur l'objet et rend le robot transparent vis-à-vis du corps de la tâche. L'objet qui est tenu par le robot est alors considéré comme un segment rigide de ce dernier. L'objectif que nous avons atteint, à ce sujet, permet d'agir de façon coordonnée sur un endroit quelconque de l'objet.

Sur le plan fonctionnel, nous avons orienté les structures synthétisées vers des implantations à base de microprocesseurs performants. La rapidité et la qualité des mouvements obtenus montrent qu'un progrès sensible a été obtenu dans l'étude et la réalisation des fonctions de coordination sur des unités de calcul rapides et peu coûteuses.

CHAPITRE II

COORDINATION EN EFFORT

II.1 - INTRODUCTION

Une première classification des systèmes-robots montre que ces machines se divisent en deux catégories. La première est caractérisée par la présence permanente ou partielle de l'homme dans le processus instantané de régulation. La seconde est caractérisée par l'absence de ce dernier vis-à-vis de ces interactions entre la machine et l'environnement. Cette dernière catégorie exige la possibilité de munir un système-robot d'une certaine autonomie voire une certaine adaptativité face à un imprévu moyennement complexe.

Des recherches avancées (31, 32) ont posé le problème d'une coordination oeil-main. D'autres ont conduit à l'étude du sens de l'effort. Ces dernières ont abouti à des résultats encourageants en élaborant des systèmes souples qui se déforment de façon à minimiser les efforts auxquels ils sont soumis. C'est ainsi que la notion de compliance (24,26) d'une structure a vu le jour.

Un système est dit compliant s'il a la possibilité de réagir en modifiant la position et l'orientation de son centre d'action de façon à minimiser les résultantes des forces et moments auxquels il est soumis. Les caractéristiques et les performances d'un tel organe se mesurent par sa généralité, son efficacité et par le domaine de grandeurs des efforts auxquels il est adapté.

Pour disposer de mécanismes compliants deux voies principales ont été explorées :

1) Structures mécaniques compliantes orientées vers des opérations d'assemblage particulières telle que l'insertion (23, 25, 26)
2) Une compliance totalement obtenue par logiciel à partir des couples moteurs ou des tensions des câbles de transmission (2, 5, 27, 28).

La première voie conduit à un mécanisme performant vis-à-vis d'une tâche principale et présente un coût relativement important. La structure compliante est considérée comme un outil que le robot doit prendre pour effectuer l'opération. On note la rapidité d'exécution et la performance vis-à-vis de la tâche à laquelle la structure compliante est destinée.

La seconde voie conduit à utiliser les informations intrinséques du robot dans le calcul des forces généralisées qui s'exercent sur sa partie terminale. Le volume de calculs mis en jeu est important. Mis à part ce dernier problème la qualité de l'information est loin d'être bonne (fortement dépendante de la qualité de la transmission) étant donné l'existence de nombreuses non linéarités (les jeux, frottements secs et la dissipation de l'énergie dans les réducteurs) dans le système de transmission. En plus l'information utile est mélangée avec les grandeurs dynamiques et statiques. Cette méthode donne des résultats acceptables lorsque la vitesse de correction est faible.

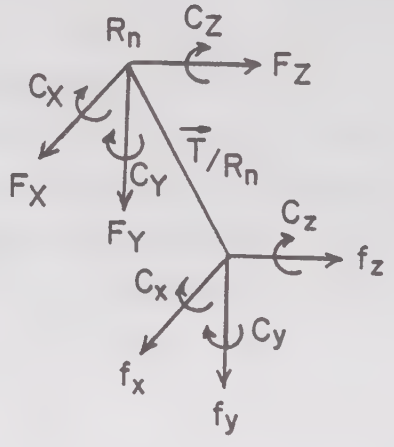
Notre démarche constitue un compromis entre ces deux voies. Elle consiste à tirer le maximum de la mécanique tout en étant générale vis-à-vis des tâches (capteur d'effort découplé mécaniquement), et à élaborer, sur la base de ce dernier, des fonctions logicielles pour assurer le contrôle en effort.

Nous présenterons à la fin de cette partie des applications aux fonctions adaptatives utilisant l'information sur les forces. Ces fonctions sont expérimentées avec la méthodologie des informations intrinsèques (les couples moteurs) et celle de notre capteur d'effort.

I.2 - COMPLIANCE ET CENTRE D'ELASTICITE

L'objet tenu par le robot subit, au cours de la tâche, une force généralisée représentée par le vecteur $f = (f_x, f_y, f_z, C_x, C_y, C_z)^t$ appliquée au point de contact, supposé unique, de ce dernier avec un autre objet. On définit le centre d'élasticité (ou centre de compliance) par le point de l'objet sur lequel s'exerce le vecteur de forces extérieures f. Ce dernier point est repéré par le vecteur $\vec{T}/R_n = (T_X, T_Y, T_Z)^t$ auquel on a placé un référentiel parallèle à celui de R_n. La réaction dans R_n est mesurée par le vecteur :

$$F = (F_X, F_Y, F_Z, C_X, C_Y, C_Z)^t$$



Il est aisé d'établir la relation entre les vecteurs F et f vis-à-vis de leurs composantes ; nous avons :

$$F_X = f_x$$
$$F_Y = f_y$$
$$F_Z = f_z$$
$$C_X = C_x + T_Z.f_y - T_Y.f_z$$
$$C_Y = C_y - T_Z.f_x - T_X.f_z$$
$$C_Z = C_z + T_Y.f_x - T_X.f_y$$

L'application d'une force (ou d'un moment) au centre d'élasticité doit conduire à des corrections de la position (ou de l'orientation) de ce dernier. Une mauvaise localisation du centre d'élasticité conduit dans le cas d'une force à des corrections de position et d'orientation ce qui va dans le sens de l'augmentation des forces de contact. Par conséquent la compliance est étroitement liée à la localisation de son centre.

Ainsi les inconnues qui doivent être spécifiées dans ces opérations sont : la localisation de ce dernier point et les amplitudes des corrections de la position et de l'orientation. Dans ce sens il est très difficile d'obtenir un comportement satisfaisant avec des structures spécialisées qui ne sont adaptées qu'à des variantes de conditions et d'opérations.

Nous nous sommes orientés vers une définition plus souple de la régulation de l'effort (compliance par logiciel adapté). Deux points importants caractérisent notre problème :

- La structure des fonctions de régulation de l'effort doit permettre une définition générale de la localisation (par le vecteur $\vec{T}$) du centre d'élasticité.

- Ces fonctions doivent effectuer les corrections de position et d'orientation respectivement au centre d'élasticité. On mesure l'intérêt des fonctions de mouvement, que nous avons développées au chapitre précédent, lorsque le référentiel d'action est translaté.

Pour cela il est utile de prélever le vecteur de forces généralisées F au niveau de la partie préhensive du robot. Ce dernier permet le calcul direct du vecteur des forces de contact f :

$$f_{(6,1)} = \left[\begin{array}{ccc|ccc} & I_3 & & & \Phi_3 & \\ \hline 0 & -T_Z & T_Y & & & \\ T_Z & 0 & -T_X & & I_3 & \\ -T_Y & T_X & 0 & & & \end{array}\right]_{(6,6)} \cdot F/_{R_n\ (6,1)}$$

où I_3 et Φ_3 sont les matrices identité et nulle d'ordre 3.

L'emploi de ces informations, dans un procédé de régulation, permet de calculer les amplitudes des corrections

de position et d'orientation du centre d'élasticité. Ces corrections peuvent s'effectuer aisément avec les fonctions de mouvements agissant sur un centre d'action translaté de $\vec{T}/R_n$ (I.5.1)

II.3 - COMPORTEMENT ET STABILITE DE LA REGULATION EN EFFORT

Le principe que nous avons utilisé (24, 36) consiste à traduire l'écart entre le vecteur de forces désiré F_d et le vecteur de force motrice F_r par une action sur le vecteur de vitesse $\dot{E}$:

$$F_d - F_r = B.\dot{E}$$

où B est la matrice de stratégie

$\dot{E}$ est le vecteur de vitesse associé au vecteur de l'effecteur E.

L'application de ce procédé à notre système de coordination conduit au schéma de régulation de base exposé sur la figure II.1. Nous étudions dans ce paragraphe le comportement et la stabilité du système face à une excitation élastique.

Un modèle dynamique simple du robot peut s'écrire au niveau de l'effecteur de la façon suivante :

$$F_r = I.\ddot{E} + Q(E)$$

où I est la matrice d'inertie équivalente dans l'espace des paramètres de l'effecteur. Nous verrons dans le 3e chapitre sa relation avec la matrice d'inertie associée à une représentation utilisant les variables généralisées.

Q(E) représente le vecteur force, dû à la pesanteur, calculé pour E.

Pour étudier le comportement de cette régulation en présence de forces extérieures, on adjoint à l'effecteur un

vecteur de forces élastiques F_{ext} :

$$F_{ext} = K.E$$

où K est une matrice de raideur.

L'équation régissant le mouvement du système articulé en boucle ouverte devient :

$$F_r = I.\ddot{E} + K.E + Q(E)$$

Lorsque la force de la pesanteur est compensée l'équation de mouvement du système en boucle fermée s'écrit :

$$F_d = I\ddot{E} + B\dot{E} + KE$$

Par ce procédé la régulation est caractérisée par un système linéaire et du second ordre. Le système converge, quand il est stable, vers l'état d'équilibre suivant :

$$F_d = F_r = KE$$

Soit V une fonction représentant l'énergie totale du système en boucle fermée :

$$V = \frac{1}{2} \{ \dot{E}^t.I.\dot{E} + E^t.K.E \}$$

où $\dot{X}^t I \dot{E}$ et $E^t KE$ représentent respectivement les énergies cinétique et potentielle. I et K sont deux matrices définies positives et symétriques.

La dérivée par rapport au temps de V décrit la forme de décroissance de l'énergie totale :

$$\dot{V} = \dot{E}^t \{ I\ddot{E} + KE \}$$

En utilisant l'équation de mouvement du système en boucle fermée nous avons :

$$\dot{V} = - \dot{E}^t.B.\dot{E}$$

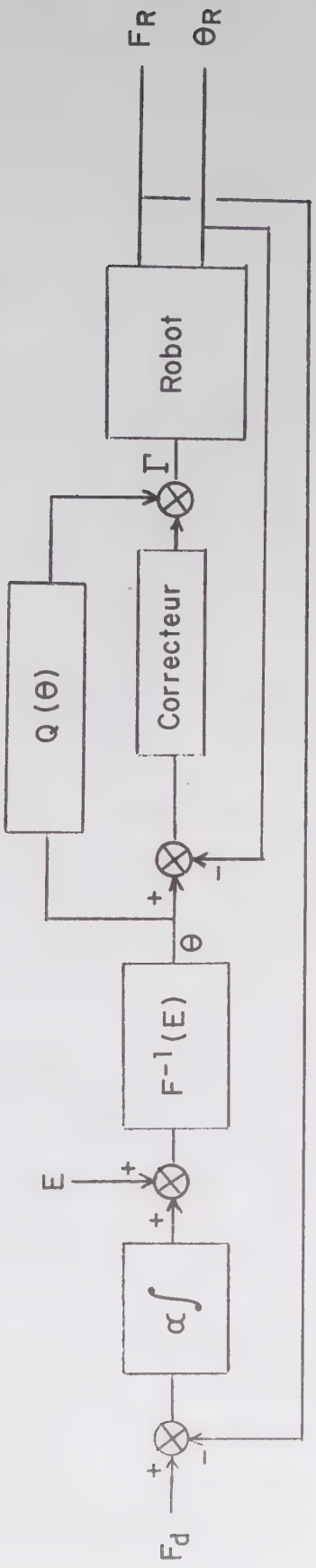


Figure II.1 : Schéma d'une régulation en effort.

Une condition suffisante de stabilité (Luapounov) est que $\dot{V}$ soit définie non positive. En d'autres termes pour que l'état d'équilibre $F_r = K.E$ soit stable il suffit que la matrice B soit définie non négative.

II.4 - CALCUL DU VECTEUR DES FORCES GENERALISEES A PARTIR DES COUPLES MOTEURS

Les éléments moteurs d'un robot peuvent (27, 29, 35) être utilisés en tant que capteurs d'effort lorsque leurs liaisons avec les articulations sont suffisamment réversibles. Cela revient dans notre calcul à lier le vecteur des forces généralisées F/R_n au vecteur des couples moteurs Γ . Le principe de l'égalité des travaux s'écrit, pour une transmission parfaitement réversible, de la façon suivante :

$$(\Delta X, \Delta Y, \Delta Z, \Delta\Phi_x, \Delta\Phi_y, \Delta\Phi_z)_{/R_n} \cdot F_{/R_n} = (\Delta \mathbf{x}^t . F)_{/R_n} = \Delta\Theta^t . \Gamma \quad \text{(II-4-1)}$$

L'équation (I-1) permet d'exprimer le vecteur d'incréments $\Delta x/R_n$ en fonction des incréments $\Delta\Theta$(cette relation est transposée dans R_n) :

$$\Delta x/R_n = H(\Theta) \cdot \Delta\Theta \qquad \text{(II-4-2)}$$

Les équations (II-4-1 et II-4-2) permettent de dégager l'expression du vecteur des forces généralisées F/R_n en fonction du vecteur des couples moteurs :

$$F/R_n = H^{-t}(\Theta) \cdot \Gamma$$

Nous exposons dans l'annexe (2) un exemple de ce calcul pour le M.A 23. Pour de nombreuses raisons ce moyen ne peut être retenu dans le cas de la programmation automatique de régulations d'effort. En effet à l'information des couples moteurs se superposent plusieurs composantes (dynamiques et statiques) autres que celles dues aux forces extérieures.

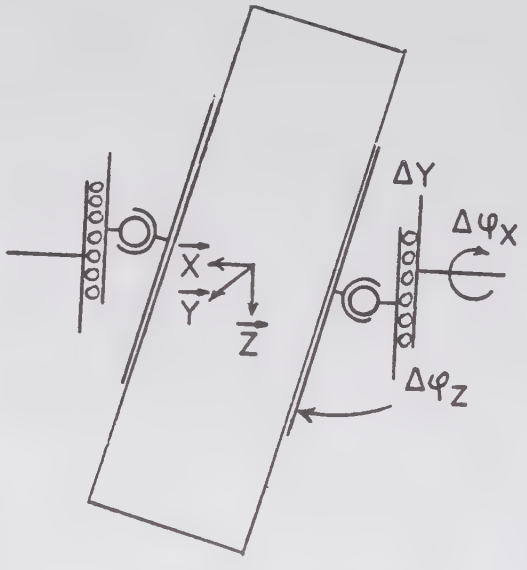
D'autre part cette information est filtrée par les fonctions de transfert globales depuis l'organe terminal jusqu'aux éléments moteurs (jeux, frottements secs et perte d'énergie dans les réducteurs). Pour ces raisons nous avons cherché à mettre au point un capteur d'effort présentant une généralité vis-à-vis de la tâche.

II.5 - OBTENTION DE L'INFORMATION A PARTIR D'UN CAPTEUR D'EFFORT

Le capteur d'effort, que nous avons conçu, est composé de deux doigts symétriques comprenant chacun trois degrés de liberté. Lorsqu'un objet est tenu entre ces deux doigts, il possède, vis-à-vis du référentiel de l'organe terminal, les trois degrés de liberté suivants :

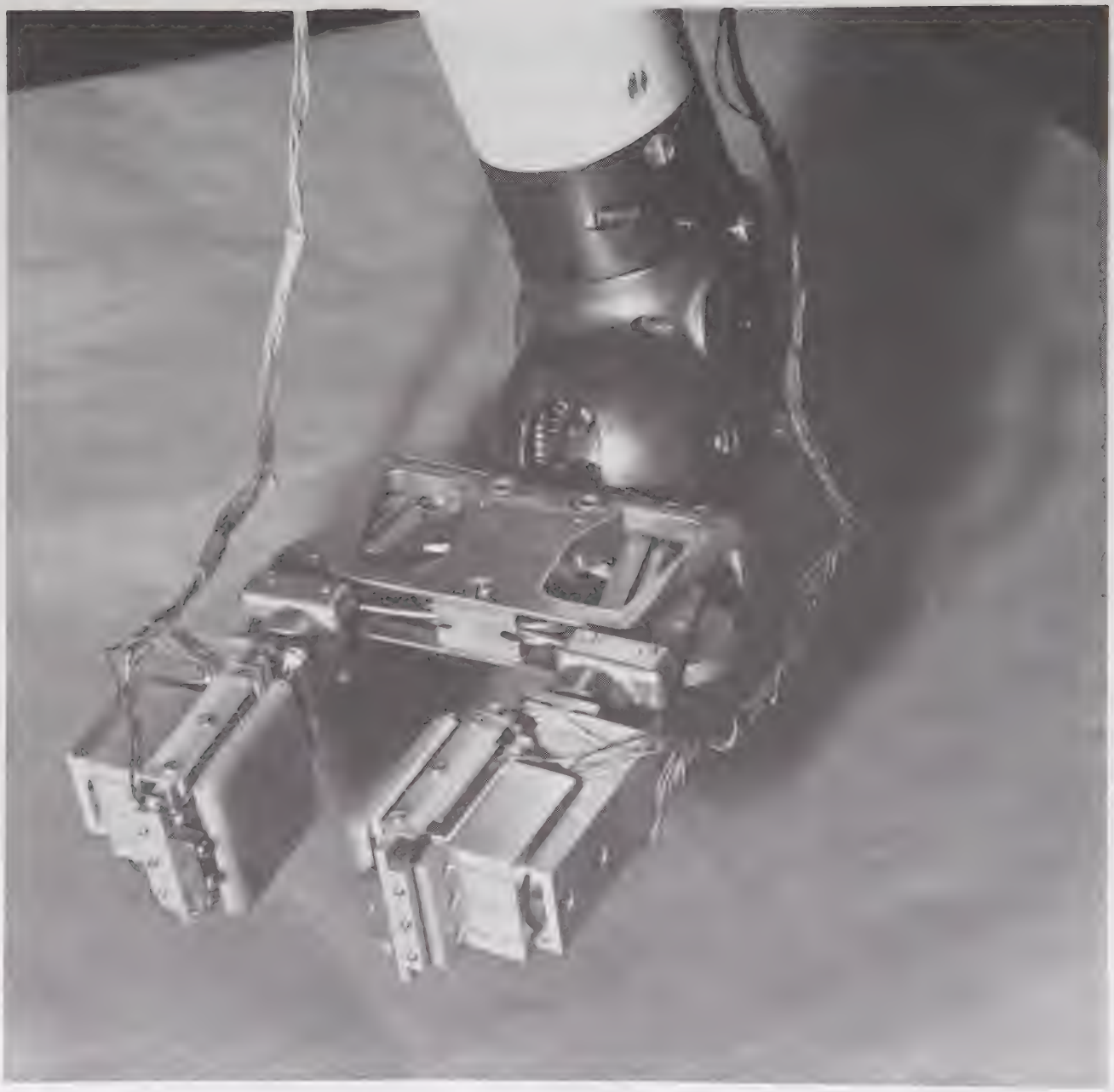
- Deux rotations (débattement de $\pm$15°) indépendantes autour des axes $\vec{X}$ et $\vec{Z}$

- Une translation selon l'axe $\vec{Y}$ (débattement de $\pm$0,8 cm).



La disposition de ces degrés de liberté est telle que le centre de prise reste fixe dans toute déformation $\Delta\Phi_X$ ou $\Delta\Phi_Z$. Ce dernier point se translate selon l'axe $\vec{Y}$. Les différents degrés de liberté sont munis de détecteurs différentiels élastiques entraînant, par leurs forces de rappel, le retour de l'objet à une position de repos (moindre effort).

Bien que muni simplement de trois degrés de liberté, la faisabilité d'opérations d'assemblage telles que l'insertion est obtenue si on élimine la part d'incertitude due à une prise largement erronée de l'objet. L'adjonction

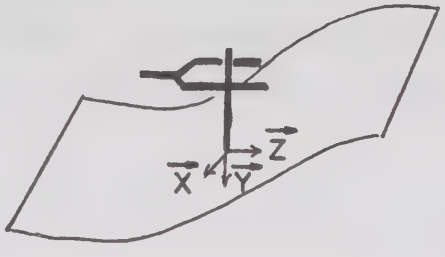


future de degrés de liberté supplémentaires (notamment les translations ΔX et ΔZ) complètera l'architecture de base de ce capteur.

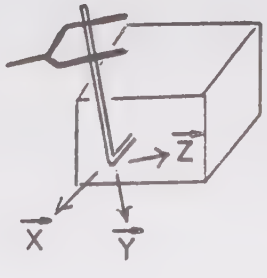
II.6 - SYNTHESE DE FONCTIONS POUR LA REGULATION DE L'EFFORT

Les fonctions de régulation que nous avons synthétisées sont caractérisées par leurs possibilités de contrôler les axes de mouvement de l'objet soit par des régulations d'effort soit en combinant ces dernières avec une commande en position. Nous précisons cette propriété sur quelques exemples simples mais significatifs :

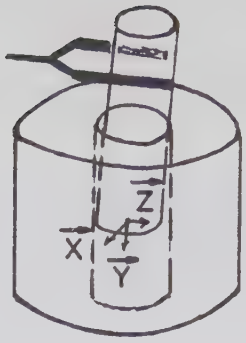
- Pour parcourir une surface non plane P avec un outil A, tenu par le robot, le mouvement de balayage de la surface est effectué par un contrôle en position selon les axes $\vec{X}$ et $\vec{Z}$. Le contact entre A et P est maintenu par une régulation en effort selon l'axe $\vec{Y}$.



- Pour tourner un volant, la rotation s'effectue autour de l'axe $\vec{X}$ du référentiel de l'effecteur translaté de $\vec{T} = (0, 1, 0)$. Mais pour assurer la bonne tenue de l'opération on régule, autour de l'effort nul, les translations selon l'axe $\vec{Y}$.



- Pour l'insertion, le contrôle des translations et des rotations correctives s'effectuent avec des régulations en effort. Ces dernières ont des consignes d'effort nul sauf selon l'axe $\vec{Y}$ qui maintient une faible pression.



La fonction prend fin lorsqu'elle aura avancé selon l'axe $\vec{Y}$ d'une longueur donnée.

Les fonctions synthétisées permettent d'effectuer aisément ces opérations. Elles se divisent en deux groupes (voir table des fonctions fig. II-2) :

1) des fonctions élémentaires variées agissant sur un ou plusieurs axes par des régulations d'effort simples (contrôle mixte = position-effort).

2) des fonctions globales avec une condition de fin d'action (objectif) spécifiée par le parcours d'une distance ou la rotation d'angles.

Elles permettent en outre de placer le centre d'élasticité de façon générale.

Nom de la fonction	Arguments	Description de l'action
REGFXYZG	F_X, F_Y, F_Z, L	Régulation générale autour des forces de consigne F_X, F_Y et F_Z avec fin d'action après parcours de L
REGCXYZG	C_X, C_Y, C_Z, Φ	Régulation générale autour des couples de consigne C_X, C_Y et C_Z avec fin d'action après rotation de Φ
REGFXYZ	F_X, F_Y, F_Z	Régulation élémentaire par rapport aux consignes F_X, F_Y, F_Z
REGFX	F_X	" " " " " " F_X
REGFY	F_Y	" " " " " " F_Y
REGFZ	F_Z	" " " " " " F_Z
REGCX	C_X	" " " " " " C_X
REGCY	C_Y	" " " " " " C_Y
REGCZ	C_Z	" " " " " " C_Z
CENTRE	T_X, T_Y, T_Z	Définition du centre d'élasticité/Rn

Fig. II-2 : Table des fonctions de régulation de l'effort

II.7 - APPLICATION AUX FONCTIONS DE MANIPULATION ET D'ASSEMBLAGE

Devant l'aspect incertain de l'environnement il n'est pas possible de décrire tous les incidents qui peuvent provoquer l'échec d'une tâche. On cherche donc à élaborer des fonctions adaptatives dont le fonctionnement permet une convergence vers un état d'équilibre souhaité.

Une tâche est généralement composée de la prise d'un objet, de son transport et enfin d'une opération particulière sur ce dernier. Nous étudierons dans ce paragraphe les fonctions de base qui permettent de faire les manipulations suivantes :

- le poser
- l'insertion et l'extraction
- la prise.

Ces fonctions ont été expérimentées (15) avec les deux méthodes, d'obtention de l'information de l'effort, exposées aux paragraphes II.4 et II.5. L'expérience est effectuée sur les deux manipulateurs cités au paragraphe I.5.3.

II.7.1 - La fonction "POSER"

Cette fonction a pour but de poser un objet, ayant une base stable, sur un support situé au-dessous de la position de départ. Lorsque l'objet entre en contact avec le support les forces calculées présentent une diminution due au fait que le support retient l'objet. En même temps le manipulateur fléchit légèrement.Tout en maintenant le contact de l'objet avec le support le système effectue des corrections de position jusqu'à l'annulation de la force qu'exerce le manipulateur sur l'objet (voir schéma fig. II-3). La fonction "POSER" présente une bonne stabilité pour des charges très variables (de quelques dizaines de g à quelques kg). La durée requise pour la convergence de la régulation est de l'ordre d'une seconde.

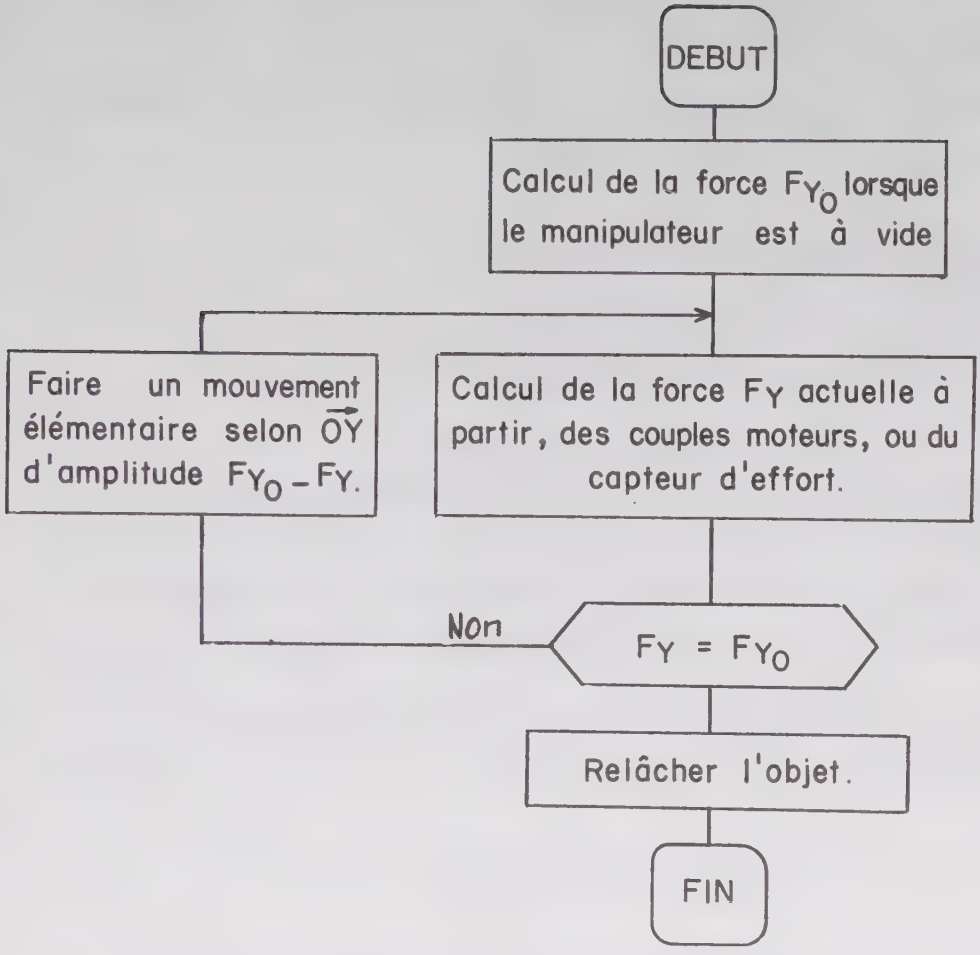


Figure II.3 : Organigramme simplifié de la fonction 'POSER'.

II.7.2 - Les fonctions "INSERER" et "EXTRAIRE"

La fonction "INSERER" permet d'introduire une tige cylindrique dans un alésage de même forme avec une faible tolérance. L'opération est composée de trois phases (voir schéma fig. II.4) :

1) Amener la tige, au contact avec la face de l'objet contenant l'alésage, par la fonction "POSER".

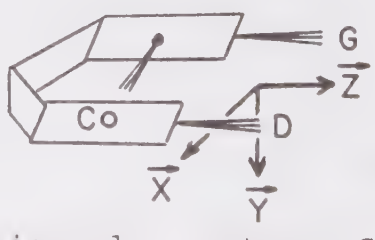
2) Effectuer une recherche en spirale, tout en maintenant une pression constante sur la tige, et détecter un changement significatif dans la régulation de pression. La tige est alors supposée coincée dans l'alésage.

3) Chercher à décoincer la tige, en faisant des corrections de positions et d'orientation en fonction des forces et couples de contact, et à progresser en profondeur. Cette phase sert également pour effectuer la fonction "EXTRAIRE".

L'accomplissement de l'insertion proprement dite (3ème phase) prend environ deux secondes.

II.7.3 - La fonction de "PRISE"

Le but de cette fonction est d'amener l'effecteur, par un processus d'affinement, à converger vers une position centrée sur l'objet et en face. Nous avons placé à cet effet des capteurs optiques (utilisés aux J.P.L. (73) et IRIA (76)), sur les deux doigts de l'effecteur. Ces derniers délivrent des informations liées grossièrement à la distance qui les sépare d'un objet réfléchissant. L'information diférentielle normalisée G-D (voir figure) permet d'effectuer les corrections de position et d'orientation de l'effecteur jusqu'au centrage de ce dernier sur l'objet (voir schéma fig. II-5). Le processus d'affinement converge généralement en 2 ou 3 secondes.



Dispositions des capteurs G,D et C

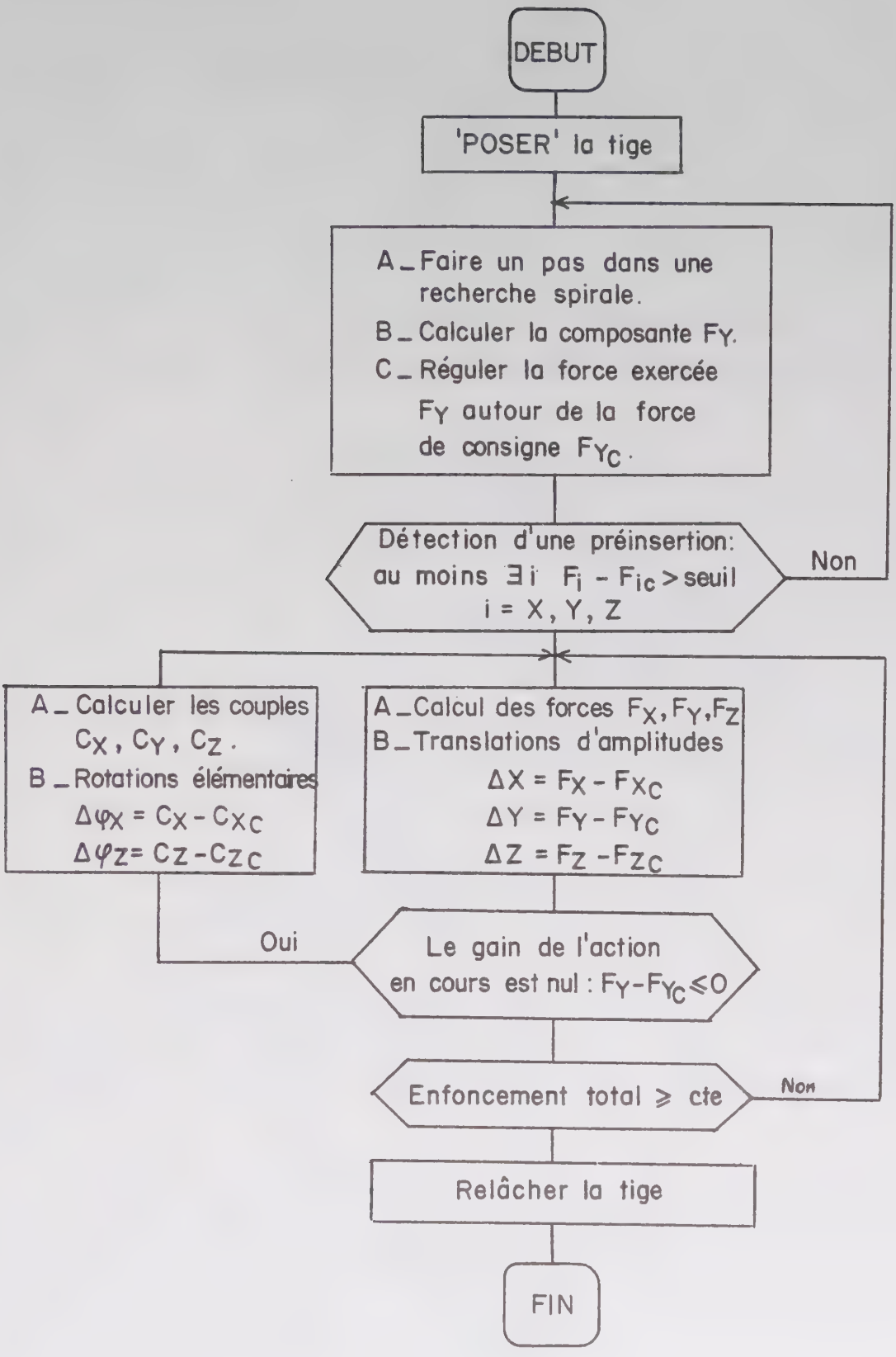


Figure II_4 : Organigramme simplifié de la fonction 'INSERER'

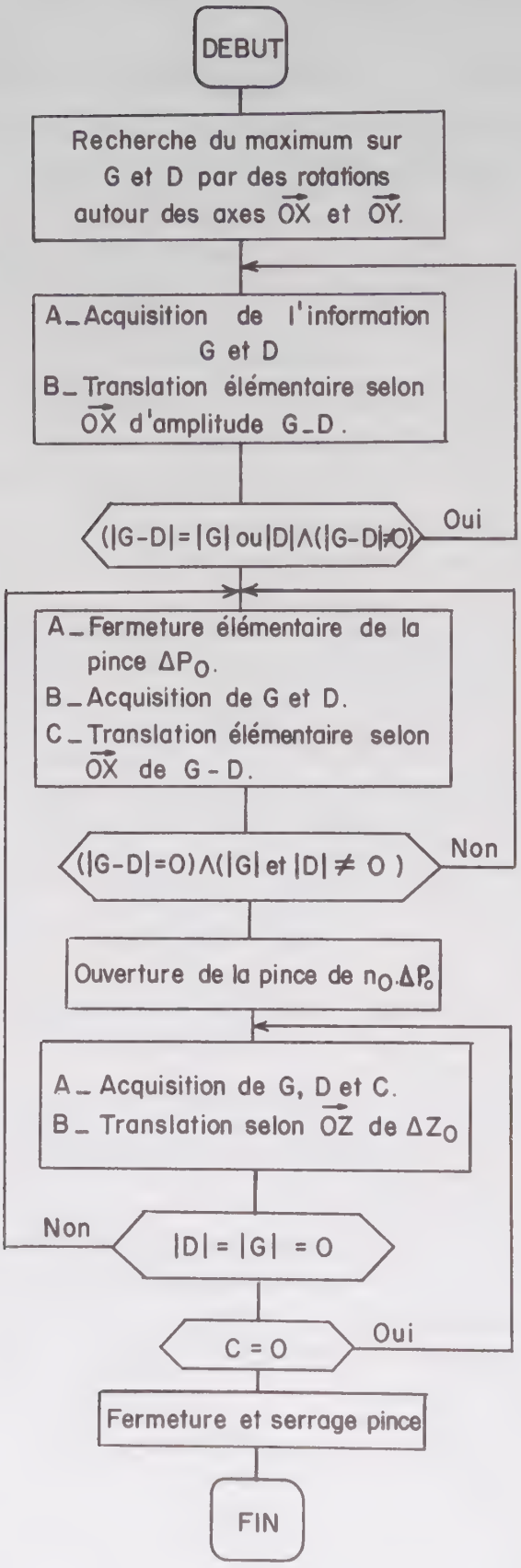


Figure II_5 : Organigramme de la fonction de 'PRISE'

II.8 - CONCLUSION

Au premier chapitre nous avons étudié et synthétisé les éléments généraux d'une coordination des mouvements. Notre objectif, à ce niveau, est de compléter ce système de commande avec une coordination en effort. Pour cela nous avons étudié plusieurs moyens pour réaliser cette fonction. En particulier nous avons conçu un capteur permettant un meilleur découplage de l'information tout en restant indépendant vis-à-vis du robot et de la tâche. Ce dernier est réalisé dans le but de constituer une liaison compliante entre l'objet manipulé et le robot.

La programmation des mécanismes d'affinement, dans les opérations d'assemblage automatique, est nettement plus simple quand on dispose de régulateurs d'effort. Ces derniers ont un emploi précis notamment en ce qui concerne leur fonctionnement par rapport au centre d'élasticité. Dans les systèmes actuels ces impératifs sont souvent à reconsidérer d'une tâche à une autre. Ainsi nous avons cherché, dans ce chapitre, à faire ressortir les éléments communs dans les opérations de régulation. Pour cela nous avons élaboré deux types de fonctions :

1) Régulations globales avec une condition de fin d'exécution
2) Régulations simples (en forces et en couples).

Leur caractéristique est de permettre des corrections de position et d'orientation vis-à-vis du centre d'élasticité qui est alors confondu, par programmation, avec le centre d'action.

Ces mécanismes, compliants par logiciel, présentent une structure simple et nécessitent peu de calculs lorsqu'ils utilisent l'information d'effort délivrée par notre capteur. Par conséquent le cycle de régulation (10 ms) est pratiquement égal à celui de la coordination d'un mouvement simple.

Nous avons enfin testé notre approche en l'utilisant dans la programmation de plusieurs fonctions adaptatives telles que INSERER, EXTRAIRE et POSER.

CHAPITRE III
COMMANDE DYNAMIQUE DES SYSTEMES ARTICULES

III.1 - INTRODUCTION

Les robots qui existent actuellement sont généralement munis de régulations simples et indépendantes assurant la fonction d'asservissement vis-à-vis d'une évolution désirée des variables généralisées. La qualité du comportement de ces robots est acceptable lorsque la vitesse ou/et la charge transportée sont faibles. Mais ce comportement s'écarte de façon sensible de celui qui est assigné lorsque cette dernière condition n'est plus remplie. Cela est dû à la nature même du système articulé pour lequel il existe des forces de couplages non linéaires.

L'intérêt d'un contrôle dynamique d'un système articulé se situe principalement par rapport aux objectifs suivants :

1. Augmentation sensible de la vitesse de manipulation notamment dans les phases de grands déplacements (transport de l'objet-outil)

2. Amélioration de la précision du robot en modes transitoire et statique (par rapport à la régulation empirique)

3. Conduire à un comportement plus stable du système articulé. Cela est réalisé avec un meilleur contrôle de l'accélération et de la décélération du mouvement. Par conséquent le matériel souffre moins, sa durée de vie est plus longue.

Pour réaliser un contrôle efficace, il est nécessaire d'analyser le comportement dynamique du système articulé.

L'étude de la nature et de l'importance des forces mises en jeu permet alors d'envisager leur compensation (synthèse à partir de l'équation temporelle) ou l'emploi de leurs coefficients dans l'élaboration d'une contre-réaction appropriée (synthèse avec une représentation d'état du système).

Nous distinguons trois problèmes liés à l'élaboration de la commande dynamique pour un système articulé :

1. Le principe d'obtention des paramètres de l'équation de comportement. Ce thème est développé aux paragraphes III.2 et III.3. Dans le premier nous exposons l'obtention de cette équation à partir du formalisme de Lagrange (le modèle mathématique). Dans le second nous proposons une méthode qui consiste à construire une représentation discrète des paramètres à partir d'une phase d'identification.

2. Le principe de calcul en ligne des paramètres pour la commande. Ce thème est développé au paragraphe III.4. Nous discuterons les problèmes de représentation, dans un calculateur, de ces paramètres en vue d'optimiser les opération de calcul.

3. Le procédé de contrôle dynamique. Ce thème est développé au paragraphe III.5. Nous discuterons les différentes approximations possibles pour le choix de la loi de commande à partir de l'équation temporelle.

Au paragraphe III.6, nous présentons une application au contrôle dynamique, avec notre approche, à un manipulateur à six degrés de liberté. Nous montrons les performances obtenues dans le cadre de cette application.

III.2 - EQUATION DE COMPORTEMENT D'UN MECANISME ARTICULE

De nombreux travaux se sont développés dans le but d'établir des modèles de comportement d'un mécanisme articulé. Plusieurs formalismes de base peuvent être utilisés pour l'obtention de tels modèles. Dans l'ordre chronologique d'apparition

ces formalismes sont :

- les équations de Newton-Euler (Hooker (38))
- les équations de Lagrange (Uicker (39) et Khalil (2))
- le principe des travaux virtuels de d'Alembert (Samin (40))
- la fonction de Gibbs (Popov (41)).

Dans le but d'établir et d'étudier l'équation de comportement nous nous sommes intéressés au formalisme de Lagrange. L'équation de mouvement d'un solide S_i de la chaîne articulée s'écrit, avec ce formalisme, de la façon suivante, (en l'absence d'énergie potentielle) :

$$\sum_{l=1}^{n} \left\{ \frac{d}{dt} \left(\frac{\partial E_{S_l}}{\partial \dot{\theta}_i} \right) - \frac{\partial E_{S_l}}{\partial \theta_i} \right\} = Q_i + \Gamma_i \quad \text{(III-1)}$$

où

n désigne le nombre de solides
E_{S_l} est l'énergie cinétique totale du solide S_l
θ_i est la ième coordonnée généralisée et $\dot{\theta}_i$ sa dérivée par rapport au temps
Q_i est la force généralisée due à la pesanteur
Γ_i est la force généralisée d'origine motrice.

Cette méthode de description possède les propriétés suivantes :

- mettre en oeuvre le calcul de l'énergie cinétique totale d'un corps dans la chaîne articulée
- permettre l'obtention de l'équation de comportement pour chacun des corps
- s'adapter à une notation matricielle ce qui rend possible son automatisation (39).

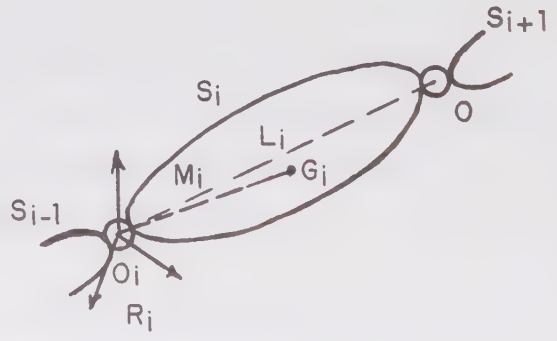
En utilisant ce formalisme W. Khalil a établi un programme général permettant le calcul littéral des paramètres

intervenant dans l'équation de comportement. Ce calcul, effectué sur ordinateur, pour un manipulateur à six degrés de liberté nécessite un volume mémoire important (200K.mots) et les résultats s'étendent sur plusieurs dizaines de pages (5).

Nous décrirons les idées principales utilisées pour l'obtention de ces équations afin de dégager les propriétés de ces paramètres. Nous utiliserons ces propriétés dans notre procédé d'estimation directe des paramètres de l'équation de comportement. Nous nous appuierons sur les notations définies par W. Khalil dans la représentation d'un solide et des vecteurs qui lui sont associés.

III.2.1 - Représentation d'un solide

Un mécanisme articulé est généralement constitué d'une chaîne de n corps reliés par n articulations de type rotoïdes ou prismatiques. La partie immobile du mécanisme est liée à un repère fixe R_o. Chacun des corps solides S_i est lié à un repère relatif R_i. Le point O_i désigne le point de liaison entre les solides S_{i-1} et S_i. M_i étant la masse de ce dernier. Le tenseur d'inertie I_i désigne la matrice d'inertie du solide S_i calculée dans le repère R_i placé au point G_i.



III.2.2 - Calcul de l'énergie cinétique

L'énergie cinétique E_{S_i} est composée des énergies cinétiques de translation et de rotation. Elle s'écrit :

$$E_{Si} = \frac{1}{2} \{ M V^t V + \omega^t I \omega \}_i$$

où V et ω sont respectivement le vecteur de vitesse de translation de G_i et le vecteur de vitesse de rotation du solide S_i par rapport à R_o et exprimés dans R_i.

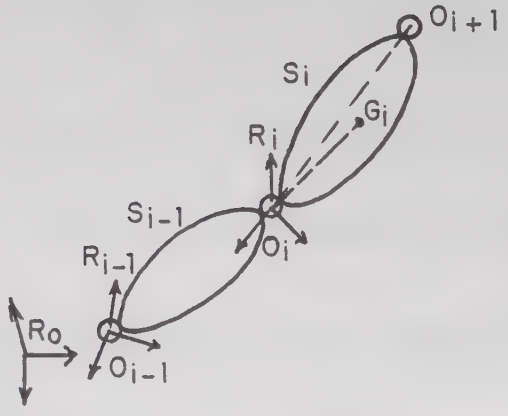
L'équation du mouvement du solide S_i s'obtient après le calcul des grandeurs suivantes :

$$V_j \ ; \ \omega_j \ ; \ E_{S_j} \qquad j : 1,\ldots,n$$

Nous rappelons les expressions de V et ω d'un solide en mouvement :

<u>La coordonnée généralisée θ_i est rotoïde</u>

Le vecteur de rotation du solide S_i par rapport au référentiel fixe R_o exprimé dans R_i est donné par la relation vectorielle :



Topologie des corps S_i

$$\omega(S_i/R_o;R_i) = \omega(S_i/R_i;R_i) + \omega(S_{i-1}/R_o;R_i)$$

avec

$$\omega(S_i/R_o;R_i) = M_i^{i-1} \ \omega(S_{i-1}/R_o;R_{i-1})$$

où M_i^{i-1} est la matrice de rotation élémentaire inverse associée à θ_i.

Le vecteur de vitesse de translation du centre de masse G_i par rapport à R_o exprimé dans R_i est donné par la relation :

$$V(G_i/R_o;R_i) = V(O_i(S_{i-1})/R_o;R_i) + V(G_i/R_i;Ri)$$

Le premier terme du membre de droite désigne le vecteur de vitesse de l'extrémité O_i du solide S_{i-1} par rapport à R_o exprimé dans R_i. Nous avons :

$$V(O_i(S_{i-1})/R_o;R_i) = M_i^{i-1} \ V(O_i(S_{i-1})/R_o;R_{i-1})$$

Par ailleurs, nous avons :

$$V(G_i/R_i;R_i) = \omega(S_i/R_o;R_i) \wedge OG_i(R_i)$$

où $OG_i(R_i)$ désigne le vecteur $\overrightarrow{O_iG_i}$ exprimé dans R_i.

Le vecteur de vitesse de translation de l'extrémité O_{i+1} par rapport à R_o exprimé dans R_i est donné par la relation :

$$V(O_{i+1}(S_i)/R_o;R_i) = V(O_i(S_{i-1})/R_o;R_i) + V(O_{i+1}(S_i)/R_i;R_i)$$

nous avons :

$$V(O_{i+1}(S_i)/R_i;R_i) = \omega(S_i/R_o;R_i) \wedge OO_{i+1}(R_i)$$

où $OO_{i+1}(R_i)$ désigne le vecteur $\overrightarrow{O_iO_{i+1}}$ exprimé dans R_i.

La coordonnée généralisée θ_i est prismatique

En conservant les mêmes définitions des vecteurs, nous avons :

$$\omega(S_i/R_o;R_i) = \omega(S_{i-1}/R_o;R_{i-1})$$

Le vecteur de vitesse du centre de masse devient :

$$V(G_i/R_o;R_i) = V(O_i(S_{i-1})/R_o;R_{i-1}) + \omega(S_{i-1}/R_o;R_{i-1}) \wedge OG_i(R_i) + V(G_i/R_{i-1};R_i)$$

Le vecteur de vitesse de l'extrémité O_{i+1} devient :

$$V(O_{i+1}(S_i)/R_o;R_i) = V(O_i(S_{i-1})/R_o;R_{i-1}) + \omega(S_{i-1}/R_o;R_{i-1}) \wedge OG_i(R_i) + V(O_{i+1}(S_i)/R_{i-1};R_i)$$

III.2.3 - Equation de mouvement

L'énergie cinétique E_{S_i} est une forme quadratique par rapport aux vitesses généralisées $\dot{\theta}_j$ (j:1,...,n). Elle admet l'écriture suivante :

$$E_{S_i} = \frac{1}{2} \overset{\circ}{\theta}^t A(i) \overset{\circ}{\theta}$$

où A(i) est une matrice définie positive et symétrique. Le développement de E_{S_i} fait apparaître des termes carrés et croisés en $\overset{\circ}{\theta}_j$:

$$E_{S_i} = \frac{1}{2} \sum_{j=1}^{i} \sum_{K=j}^{i} a_{jK}(i) \overset{\circ}{\theta}_j \overset{\circ}{\theta}_K$$

avec

$$a_{jK}(i) = 2 (A(i))_{j,K} \quad \text{pour } j \neq K$$

$$a_{jj}(i) = (A(i))_{j,j}$$

D'après l'équation III.1 nous avons pour le solide S_i :

$$\sum_{l=1}^{n} \frac{d}{dt} \left(\frac{\partial E_{S_l}}{\partial \overset{\circ}{\theta}_i} \right) = \sum_{j=1}^{n} A_{ij} \overset{\circ\circ}{\theta}_j$$

Ce terme représente la force généralisée due aux effets inertiels appliqués sur le solide S_i. Les paramètres A_{ij} sont fonctions de la configuration du mécanisme et du dimensionnement massique et métrique de ce dernier. En particulier ces termes A_{ij} dépendent des variables de configuration $\theta_{i+1}, \ldots, \theta_j, \ldots, \theta_n$.

Le calcul du second terme de l'équation III-1 consiste à effectuer la dérivation partielle des termes A_{ij} par rapport à la variable généralisée θ_l :

$$\frac{\partial E_{S_i}}{\partial \theta_l} = \frac{1}{2} \overset{\circ}{\theta}^t \left\{ \frac{\partial}{\partial \theta_l} A(i) \right\} \overset{\circ}{\theta}$$

Le groupement de ces termes selon les vitesses croisées $\overset{\circ}{\theta}_j \overset{\circ}{\theta}_K$ donne la forme finale suivante :

$$- \sum_{l=1}^{n} \frac{\partial E_{S_l}}{\partial \theta_i} = \sum_{j=1}^{n} \sum_{K=j+1}^{n} B_{ijK} \, \dot{\theta}_j \dot{\theta}_K + \sum_{j=1}^{n} C_{ij} \, \dot{\theta}_j^2$$

Les termes du second membre de cette équation désignent respectivement les forces de Coriolis et centrifuge appliquées au solide S_i. L'équation de mouvement du solide S_i est obtenue à partir de ces trois forces généralisées. Sous forme matricielle l'équation globale de la chaîne articulée s'écrit :

$$A_{n,n} \, \ddot{\theta} + B_{n,L} \, \ddot{\theta\theta} + C_{n,n} \, \dot{\theta}^2 = \Gamma_{n,1} + Q_{n,1}$$

où A est la matrice des coefficients d'inertie
B " " " " " des forces de coriolis
C " " " " " " " centrifuges

$$\ddot{\theta} = \frac{d^2}{dt^2} \theta$$

$$\ddot{\theta\theta} = [\theta_1\theta_2, \ldots, \theta_1\theta_n, \theta_2\theta_3, \ldots, \theta_2\theta_n, \ldots, \theta_{n-1}\theta_n]^t$$

$$\dot{\theta}^2 = [\theta_1^2, \ldots, \theta_n^2]^t$$

$$L = n.(n-1)/2$$

Une application de ce calcul à un manipulateur à six degrés de liberté est présentée en annexe 4.

III.2.3 - Conclusion

Nous avons vu que le modèle mathématique conduit au calcul littéral des forces qui régissent le mouvement du système articulé. Ces dernières sont déterminées par la connaissance des paramètres qui y figurent. Ces paramètres sont nombreux, leurs expressions sont complexes (4). Nous avons vu dans l'introduction de ce paragraphe qu'il existe des systèmes

de calcul littéral spécialisés pour le calcul de ces paramètres. Mais la forme de ces derniers à la sortie de ces systèmes reste complexe et nécessite de nombreux aménagements. Elle se prête mal aux simplifications que l'ingénieur, chargé de réaliser la commande, pourrait effectuer. Par exemple nous pourrons avoir à éliminer une ou plusieurs variables généralisées dans l'expression d'un paramètre quand la longueur et la masse du segment correspondant est relativement faible devant le reste du système (faible influence sur le paramètre).

Ainsi nous pensons que la solution envisageable, dans le cadre du calcul littéral des paramètres, se situe au niveau de l'existence d'un interpréteur puissant (par exemple type APL) spécialisé dans le calcul littéral pour la robotique. Outre l'élaboration automatique des paramètres à partir des données massiques et métriques (ce qui est le cas des systèmes actuels) il doit disposer de procédures puissantes orientées vers les opérations suivantes :

1. Simplifications des termes redondants et groupement selon les variables dépendant de la configuration (les termes en cos (θ_i), sin (θ_i), etc.)

2. Elimination de certaines variables qui jouent un rôle peu important dans un paramètre en particulier, ensuite réutilisation de l'opération 1.

3. Etant donné l'ensemble des paramètres (simplifiés) déduire l'ordonnancement optimal des opérations de calcul de ces derniers (pour obtenir la formule à appliquer en temps réel). La forme obtenue doit, en particulier, présenter un paramètre sous la forme d'une somme de produits de termes constants (fonction des facteurs métriques et massiques) et des termes qui sont fonction de la configuration.

4. La possibilité de transformer une chaîne de caractère (un paramètre) en expression exécutable en vue d'étudier (avec une définition donnée des facteurs métriques et mas-

siques) l'influence du paramètre par rapport aux forces globales dans lesquelles il figure.

Mis à part ces problèmes de représentation la fidélité de la modélisation n'est pas toujours démontrée étant donné la complexité de la structure mécanique. En effet, un segment du système articulé est, en général, une pièce mécanique creuse (traversée par les câbles de transmissions) qui n'admet pas toujours un axe de symétrie.

En raison de ces difficultés nous proposons une méthode plus directe pour approcher ces paramètres. Nous développons cette dernière dans le paragraphe suivant.

III.3 - OBTENTION DIRECTE DE PARAMETRES DE L'EQUATION DE COMPORTEMENT

Nous avons vu dans le paragraphe précédent la formulation mathématique de l'équation de comportement d'une chaîne articulée. Cette dernière conduit, avec le formalisme de Lagrange, à l'écriture des coefficients dynamiques en fonction des données structurales, massiques et métriques d'une chaîne articulée.

L'obtention directe de ces coefficients consiste à leur identification sur la base d'une modélisation simple décrivant l'interaction entre l'organe moteur et l'articulation à travers le système de réduction. Elle s'adresse essentiellement aux systèmes munis de transmissions réversibles.

Cette méthode, que nous proposons, présente les points d'intérêt suivants :

- Tenir compte des facteurs et paramètres réels qui se manifestent dans le comportement. Et pouvoir ainsi suivre leurs évolutions au cours du vieillissement du matériau qui compose le robot.

- S'orienter vers des techniques générales de tabulation rendues

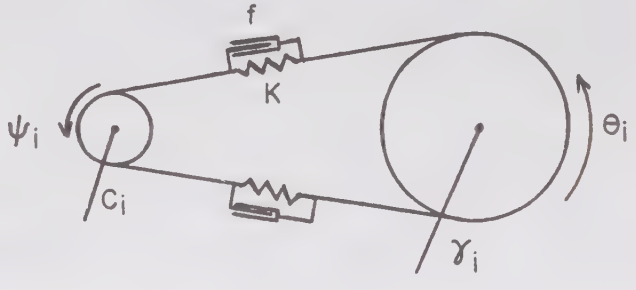
envisageables par l'évolution incessante de la technologie, permettant ainsi l'élaboration d'algorithmes de commande compatibles avec les exigences du temps-réel.

Nous précisons que cette méthode ne permet d'identifier que les paramètres liés aux forces dynamiques principales. Ces dernières sont souvent les forces d'origine inertielles (2, 4, 5). Ainsi nous nous intéressons aux paramètres d'inertie ressentis par les organes moteurs.

III.3.1 - Modélisation de la chaîne : organe moteur-réducteurs - articulation.

L'identification des paramètres de l'équation de comportement nécessite une connaissance suffisamment fine d'un modèle de transfert de l'effort depuis l'organe moteur jusqu'à l'articulation. Une étude (42) a été menée au L.A.M. dans le but de dégager un modèle de comportement d'une chaîne comprenant un organe moteur, un système de réduction et une articulation munie d'une charge. Cette étude a mis en évidence l'existence de 44 variables généralisées. L'ordre du système a été réduit à 24 en ne considérant que les mouvements antisymétriques qui affectent les variables d'entrée et de sortie. Cette étude a le mérite de proposer un modèle simplifié en se limitant au mode rigide et au premier mode élastique.

Nous avons posé des hypothèses restreintes portant sur l'importance relative des frottements visqueux intervenant dans ce modèle.



ψ_i : angle moteur pour le ième élément

θ_i : angle articulation

C_i : couple moteur

γ_i : couple sur l'articulation

f : frottement visqueux dans les réducteurs

K : coefficient d'élasticité.

Schéma de principe de la transmission

Le modèle qui se dégage de ces considérations est simple (annexe 3) et est exploitable sur le plan pratique. Il a été validé pour les deux robots sur lesquels nous avons testé cette approche.

III.3.2 - Principe de l'identification et ordonnancement

Pour pouvoir appréhender les différents éléments de la matrice d'inertie A il est nécessaire de relever les observations suivantes :

1. Le coefficient d'inertie A_{ij} ($i \leqslant j$; i : 1,...,n) dépend de la configuration aval à la ième articulation, c'est-à-dire des variables généralisées $\Theta_{i+1},\ldots,\Theta_n$.

2. La fonction $H^{i}_{C\psi} = \frac{\psi_i}{C_i}$ (avec la représentation de Laplace) représente la fonction de transfert entre le couple moteur C_i et l'angle moteur ψ_i pour le ième élément de la chaîne. Le seul paramètre d'inertie intervenant dans cette fonction est l'élément $A_{ii}(\Theta)$ de la matrice A.

3. La fonction $H^{i,j}_{C\Theta} = \frac{\Theta_j}{C_i}$ représente la fonction de transfert entre le ième couple moteur C_i et le jième angle d'articulation. Elle est fonction des éléments $A_{ii}(\Theta)$, $A_{jj}(\Theta)$ et $A_{ij}(\Theta)$ de la matrice A.

4. Une procédure d'identification $I(C,\psi/H_{C\psi})$ doit tenir compte des forces de la pesanteur ainsi que de leurs évolutions lors du mouvement de l'angle ψ dû au couple moteur d'excitation C.

Ainsi pour identifier les éléments A_{ii}, $A_{i,i+1},\ldots,A_{i,n}$ il suffit d'appliquer un couple approprié C_i et de relever les évolutions des angles moteurs qui en résultent : $\Phi_i,\Phi_{i+1},\ldots,\Phi_n$. Le couple d'excitation C_i est appliqué sur le ième arbre moteur, les autres éléments de la chaîne sont en équilibre (en boucle ouverte) sur la position configuration Θ . Une procé-

dure d'identification $I(C_i, \Phi_i / H^i_{C\psi})$ permet alors de dégager le terme A_{ii} sur la base du modèle $H^i_{C\psi}$. D'autre part la procédure $I(C_i, \Phi_j / H^{ij}_{C\psi})$ permet de dégager le terme A_{ij} sur la base du modèle $H^{ij}_{C\psi}$ (A_{ii}, A_{jj}).

Par conséquent le meilleur ordonnancement de ces opérations, pour l'identification complète de A (θ), est le suivant :

Couples imposés	Evolutions relevées	Paramètres identifiés
C_n	ψ_n	$A_{n,n}$
C_{n-1}	ψ_{n-1} ; ψ_n	$A_{n-1,n-1}$; $A_{n-1,n}$
C_2	ψ_2 ; ψ_3 ; ... ; ψ_n	$A_{2,2}$; $A_{2,3}$; ... ; $A_{2,n}$
C_1	ψ_1 ; ψ_2 ; ..... ; ψ_n	$A_{1,1}$; $A_{1,2}$; .... ; $A_{1,n}$

III.3.3 - Lissage d'une fonction multivariable

Le paramètre $A^*(\theta)$ identifié est représenté dans une table de correspondance vis-à-vis des valeurs discrètes de la configuration θ.

Nous admettons que la fonction multivariable A (θ) possède une décomposition unique sur une base de q fonctions orthonormales $\phi_i(\theta)$:

$$A(\theta) = \sum_{i=1}^{q} a_i \, \phi_i \, (\theta) \quad \text{avec} \quad \theta = (\theta_1, \ldots, \theta_n)^t$$

Les valeurs estimées $A^*(\theta)$ s'écrivent en fonction de $A(\theta)$ et d'une erreur ε :

$$A^* (\theta) = A (\theta) + \varepsilon$$

Soit R un critère quadratique sur l'erreur ε . Les coefficients a_i (i:1,...,q), qui minimisent le critère R, s'obtiennent aisément :

$$a_i = \int_\Theta \phi_i (\theta) A (\theta) d\theta$$

Les fonctions multivariables $\phi_i(\theta)$ sont engendrées à partir de fonctions monovariables orthonormées comme les polynomes de Legendre. Nous rappelons l'équation récursive donnant les polynomes $P_{K+1}(\theta_j)$ orthogonaux d'ordre K+1 :

$$(K+1)P_{K+1} (\theta_j) - (2K+1)\ \theta_j\ P(\theta_j) + K\ P_{K-1} (\theta_j) = 0 \text{ pour } K \geqslant 1$$

Les polynomes orthonormaux $P_K^* (\theta_j)$ se déduisent des $P_K(\theta_j)$ par la relation (Hilbert) :

$$P_K^* (\theta_j) = \sqrt{\frac{2K + 1}{2}}\ P_K (\theta_j)$$

III.3.4 - Application au manipulateur M.A.23

L'étude sur le manipulateur M.A.23 (25 kg) est effectuée pour les quatre premiers degrés de liberté. Chacun de ces degrés de liberté est discrétisé sur 10 niveaux. Un paramètre A_{ij} est fonction au plus de deux variables généralisées principales.

L'identification des éléments diagonaux A_{ii} de la matrice A s'effectue avec les modèles de transfert $H^i_{C\psi}$ qui sont simples. En effet, l'erreur relative maximale, par comparaison aux valeurs des A_{ii} obtenues à partir du modèle mathématique, est de l'ordre de 4 %. Ce bon résultat n'a pu être obtenu qu'en respectant les points suivants :

- Les paramètres du modèle $H^i_{C\psi}$ ont été identifiés avec la plus grande précision.

- La forme d'excitation C(t) est choisie de façon à minimiser l'effet des élasticités sur l'évolution de $\psi(t)$ (voir annexe 3).

- Les couples d'excitation sont importants, par conséquent l'effet néfaste des frottements secs, sur $\psi(t)$, est très réduit.

Les erreurs relatives maximales observées sur les estimées A_{ij} ne sont pas de même ordre de grandeur que celles calculées pour les termes A_{ii}. Ces erreurs sont importantes * et vont jusqu'à 11 %. Comme nous l'avons vu au paragraphe III.3.2 les termes extradiagonaux A_{ij} s'estiment sur la base de la connaissance des termes diagonaux A_{ii} et A_{ij} qui sont eux-mêmes entachés d'erreurs. En plus le couple C_j de couplage inertiel, dû à l'apparition de $\ddot{\psi}_i$, appartient au domaine faible ($\leqslant 0.1$ du couple maximal), par conséquent il est sensible aux frottements secs.

Pour rendre exploitables les estimées A_{ij}, nous avons employé des procédures de lissage pour les fonctions multivariables. Nous avons utilisé cette technique (III.3.3) avec les polynomes orthonormés de Legendre. Des tests sur les fonctions $A_{ij}(\theta)$, obtenues par le modèle mathématique, nous ont montré que l'approximation à l'ordre 3 (pour les polynomes orthonormés monovariables) reste suffisante. L'erreur relative maximale obtenue, après lissage, pour les paramètres A_{ij} est de l'ordre de 7 %.

---

* Les conséquences des erreurs relatives faites dans l'estimation de A_{ii} ou A_{ij} ne sont pas les mêmes sur les mouvements quels que soient ceux-ci. Ainsi, 11 % sur un paramètre A_{ij} peut être meilleur ($A_{ij}\ \ddot{\theta}_j / A_{ii}\ \ddot{\theta}_i$) que 4 % sur un A_{ii} du point de vue des conséquences.

III.3.5 - Conclusion

Nous avons étudié aux paragraphes III.2 et III.3 l'obtention des paramètres de l'équation de comportement pour un système articulé. Au paragraphe III.2 nous avons exposé la méthode généralement suivie pour le calcul littéral de ces paramètres à partir d'un formalisme mathématique basé sur la modélisation des segments qui composent le système articulé. Nous avons exposé les problèmes liés à la fonctionnalité de cette méthode dans la mise en oeuvre de la commande dynamique. Au paragraphe III.3 nous avons proposé une méthode qui consiste à approcher les paramètres, de l'équation de comportement, par des considérations sur le fonctionnement réel du système. Notre approche tente d'approfondir et d'exploiter les relations réelles qui existent entre l'actionneur et l'articulation dans le but de synthétiser une loi de commande plus performante.

Une comparaison succincte nous montre l'apport que présente notre approche par rapport à celle du modèle.

	Modèle mathématique des paramètres	Identification directe des paramètres
Obtention	Programmes d'élaboration automatique des expressions littérales des paramètres	Phase d'apprentissage à partir de l'analyse des effets réels sur le système articulé
Calcul pour la commande en ligne	1. Calcul de l'expression du paramètre ou 2. Discrétisation, tabulation et interpolation	1. Représentation discrète, tabulation et interpolation ou 2. Calcul d'expressions polynomiales synthétisées
Avantages	1. Possibilité d'automatiser son calcul littéral 2. S'applique aux robots munis de transmissions élastiques ou non	1. Possibilité d'automatiser la procédure d'identification 2. Suivre le vieillissement du matériau mécanique 3. Fidélité de la représentation par rapport à la mécanique
Inconvénients	1. Expressions complexes et longues 2. Nécessité de procédures puissantes de simplification et de groupement 3. La fidélité du modèle n'est pas totalement prouvée	Dépend de la qualité de la réversibilité et par conséquent se limite aux systèmes réversibles

III.4 - CALCUL EN LIGNE DES PARAMETRES DE LA DYNAMIQUE

Les paramètres de l'équation de comportement sont fonction de la configuration des variables généralisées. La commande conduit au calcul, en ligne, de ces paramètres en fonction de la configuration actuelle du manipulateur. Nous avons vu que le modèle mathématique fournit les expressions analytiques de ces paramètres. Notre procédé d'identification fournit les valeurs de ces derniers pour des positions discrètes du vecteur Θ. Nous discuterons dans ce paragraphe la façon de structurer le calcul sous ces deux aspects : représentations analytique et discrète.

III.4.1 - Optimisation avec la représentation analytique des paramètres

Un paramètre P, de l'équation de comportement, peut s'écrire sous la forme suivante :

$$P = a_o + \sum_{i}^{K} a_i . T_i$$

où $a_o, a_1, \ldots, a_K$ sont des fonctions d'une part des données massiques, inertielles et métriques du robot et d'autre part de la charge portée par ce dernier

T_i est une fonction de produit d'un certain nombre de $\cos(\theta_j)$ et de $\sin(\theta_j)$ $(j : 1, \ldots, n)$

Vu sous l'angle d' "Algorithme de commande" le paramètre P est fonction de plusieurs niveaux de dépendance :

1. Dépendance structurelle : l'architecture du robot fixe la forme analytique générale du paramètre P

2. Dépendance matérielle intrinsèque : pour deux robots ayant une même architecture leurs paramètres P diffèrent par les termes $a_o, \ldots, a_K$, mais les éléments $T_I, \ldots, T_K$ sont les mêmes.

3. Dépendance avec la charge : tant que la charge n'a pas changé les éléments $a_o,\ldots,a_K$ sont constants quel que soit θ.

4. Dépendance avec la configuration : les termes $T_o,\ldots,T_K$ dépendent de θ.

On remarque qu'un niveau de dépendance englobe tous ceux qui le suivent.

Dans le cadre de notre application, à la commande par le modèle analytique des paramètres, nous avons matérialisé ces différents niveaux de dépendances par des structures modulaires. Deux caractéristiques importantes sont alors obtenues.

La première concerne l'optimisation des opérations de calcul. Cela revient à écrire un paramètre selon la forme décrite plus haut. Pendant le mouvement on calculera uniquement les termes $T_1,\ldots,T_K$. La modification de la charge opère sur la reconfiguration des termes $a_o,\ldots,a_K$ qui demeurent constants pendant le mouvement.

La seconde concerne la généralité de l'algorithme de commande. Quand on change de robots, avec la même architecture, seul le niveau 2 est modifié (cas des manipulateurs M.A 23 10 kg, 25 kg et M.A 23M).

A titre d'exemple, le calcul des coefficients d'inertie principaux (10 paramètres) et du vecteur de la pesanteur requiert 6 ms avec cette représentation sur notre unité à microprocesseur.

III.4.2 - Optimisation avec la représentation discrète des paramètres

Disposant des valeurs discrètes de la matrice A (θ) il est possible d'envisager deux moyens pour leur exploitation en temps réel.

Le premier moyen revient à approcher un paramètre

par une fonction polynomiale P(θ) (III.3.3). La procédure consiste à calculer, en ligne, cette dernière fonction pour les valeurs de θ courantes. Pour obtenir une précision de 2 ou 3 % le nombre d'opérations arithmétiques est généralement grand. Pour fixer les idées, un paramètre P (θ) fonction de deux variables généralisées (dans $[-\Pi, +\Pi]$) nécessite, au moins, une approximation polynomiale à l'ordre 4 pour obtenir une erreur relative globale de quelques %. Dans ce cas le nombre d'opérations arithmétiques, qui doit s'effectuer en ligne, pour le calcul de P (θ) s'élève à ≃ 50 multiplications et additions.

Le second moyen consiste à tabuler les valeurs discrètes de P (θ). Cela permet de diminuer notablement l'ordre du polynome d'interpolation et par conséquent le nombre d'opérations arithmétiques. Ce problème a été traité par Raibert (43) et Khatib (6).

Raibert propose un découpage de l'espace des variables généralisées en zones dans lesquelles le paramètre P (θ) reste constant. Ainsi, avec une discrétisation à 8 niveaux, la mémorisation d'un paramètre (fonction de 6 variables) requiert 4 K mots de 12 bits. Bien que la taille mémoire puisse trouver une solution avec le développement de la technique l'inconvénient de cette idée réside dans le fait qu'elle traite les variables généralisées avec une importance égale.

En ce qui nous concerne nous pensons qu'il est inutile, pour les paramètres d'inertie au moins, d'aller au-delà d'une précision voisine de 5 %. D'autre part comme nous l'avons observé dans la pratique : un paramètre P (θ) est quasiment fonction de deux variables généralisées principales lorsque le dimensionnement métrique et massique de l'organe terminal est faible devant celui des trois premiers degrés de liberté (ce qui est le cas pour la plupart des robots manipulateurs). Dans ces conditions l'approximation polynomiale à l'ordre 3, pour le paramètre P (θ), nécessite 15 multiplications et 15 additions lorsque 10 niveaux de discrétisation sont considérés.

III.5 - PROCEDES DE CONTROLE EN DYNAMIQUE

Nous avons vu (III.2.3) qu'une structure articulée rigide est régie par l'équation de comportement suivante :

(III.5.1)

$$\underset{\text{forces inertielles}}{A.\ddot{\Theta}} + \underset{\text{de Coriolis}}{B.\dot{\Theta}\dot{\Theta}} + \underset{\text{Centrifuge}}{C.\dot{\Theta}^2} = \underset{\text{extérieurs}}{\Gamma_{ext}} + \underset{\text{de gravité}}{Q}$$

Elle montre que le système est régi par n équations différentielles du second ordre, non linéaires et couplées. Ainsi plusieurs chercheurs (44, 45, 46, 2, 47) ont proposé des procédés de contrôle pour rendre l'équation globale du comportement découplée et linéaire.

Le problème de la commande, par la linéarisation de l'équation (III.5.1) a été étudié par KHAN (47) qui proposa un contrôle "bang-bang". Outre les difficultés pratiques que soulève un tel mode, la linéarisation du système est difficile à admettre.

Le principe du découplage peut s'effectuer directement sur l'équation temporelle (III.5.1). Ainsi des solutions ont été élaborées dans le domaine des variables généralisées (44, 45, 2) et d'autres dans le domaine d'une représentation de l'effecteur (6).

Des solutions équivalentes (2) sont obtenues en appliquant la théorie du découplage non linéaire à une représentation d'état ou système (46) représenté par l'équation III.5.1.

Nous présentons dans ce paragraphe le principe général du découplage appliqué sur l'équation temporelle. Nous déduirons les expressions de la grandeur de commande réalisant le découplage au niveau des variables généralisées et au niveau d'une représentation de l'effecteur. Nous discuterons ensuite les différentes approximations du problème.

III.5.1 - Découplage au niveau des variables généralisées

Le principe du découplage consiste à définir un mode de contrôle pour lequel chaque degré de liberté est régi, en boucle fermée, par une équation différentielle linéaire, du second ordre et à coefficients constants. L'équation globale du système s'établit avec l'hypothèse de découplage :

$$\ddot{\varepsilon} + GH.\dot{\varepsilon} + G.\varepsilon = 0 \quad \text{(III.5.2)}$$

où $\varepsilon(t) = \Theta_d(t) - \Theta(t)$ vecteur d'erreur de position

$\Theta = (\Theta_1, \ldots \Theta_n)^t$ vecteur des coordonnées généralisées

$\Theta_d = (\Theta_{d1}, \ldots \Theta_{dn})^t$ vecteur des consignes

GH est la matrice diagonale des amortissements élémentaires

G est la matrice diagonale des gains élémentaires.

La matrice des coefficients d'inertie A étant définie positive donc régulière, nous avons pour l'équation (III.5.2) :

$$A.\ddot{\varepsilon} + A.GH.\dot{\varepsilon} + A.G.\varepsilon = 0$$

En introduisant cette dernière équation dans l'équation du modèle (III.5.1) nous avons :

$$A\ddot{\Theta}_d + A.GE.\dot{\varepsilon} + A.G.\varepsilon = \Gamma + Q(\Theta) - B.\ddot{\Theta}\ddot{\Theta} - C.\dot{\Theta}^2$$

Pour simplifier les notations posons :

$$\Gamma_{NL} = B.\ddot{\Theta}\ddot{\Theta} + C.\dot{\Theta}^2 - Q(\Theta)$$

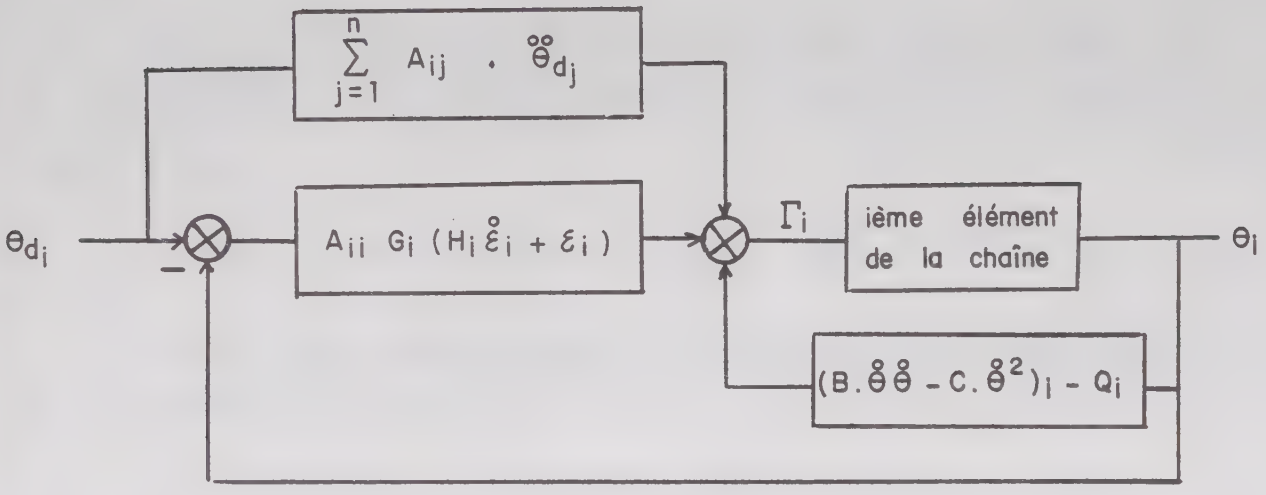
L'équation du couple moteur Γ s'exprime de la façon suivante avec ce procédé :

$$\Gamma = A.\ddot{\Theta}_d + A.GH.\dot{\varepsilon} + A.G.\varepsilon + \Gamma_{NL}$$

Au niveau du ième organe moteur nous avons :

$$\Gamma_i = \Gamma_{NLi} + A_{ii} G_i \{H_i . \dot{\varepsilon}_i + \varepsilon_i\} + \sum_{j=1}^{n} A_{ij} . \ddot{\theta}_{dj} \qquad \text{(III.5.3)}$$

D'où la structure de la fonction de contrôle au niveau du ième élément du système :



Les caractéristiques de ce procédé sont les suivantes :

1. Injection des accélérations des consignes pondérées par les coefficients d'inertie correspondants

2. Les gains et les amortissements sont pondérés par les éléments diagonaux A_{ii} donc dépendants de la configuration θ et de la charge.

3. Compensation des forces non linéaires $(B.\dot{\theta}\ddot{\theta} - C.\dot{\theta}^2)$ et $Q_i(\theta)$.

Plusieurs approximations ont été étudiées dans le cadre de ce principe de découplage. Ce fut Paul (44) en 1973

qui proposa un principe de découplage proche de ce dernier (équation III.5.3). Il s'intéressa aux termes diagonaux de la matrice A et négligea les forces centrifuges et de coriolis. L'expression de sa grandeur de commande est la suivante :

$$\Gamma_i = A_{ii} G_i \, (H_i . \overset{\circ}{\varepsilon}_i + \varepsilon_i) + A_{ii} \overset{\circ\circ}{\theta}_{di} + Q_i$$

Cette commande fut appliquée au "Sheinman-arm" pour lequel elle donne des résultats satisfaisants.

BEJCZY (45) proposa un contrôle qui consiste à laisser invariants les gains et les amortissements de la régulation. La loi de commande s'écrit :

$$\Gamma_i = G_i (H_i \overset{\circ}{\varepsilon}_i + \varepsilon_i) + \sum_{j=1}^{n} A_{ij} \overset{\circ\circ}{\theta}_{dj} + Q_i$$

Ce dernier suppose que la matrice d'inertie varie peu ce qui n'est pas justifié lorsque le système fonctionne en charge.

Liégeois et Khalil (2) étudièrent en 1978 cette loi sur un modèle de simulation (également en mode hors ligne) et montrèrent les performances du découplage lorsque ce principe est appliqué dans sa totalité (commande prédictive). Il est certain que cette démarche donne théoriquement de meilleurs résultats mais la rapidité des calculateurs actuels ne permet pas son implantation en ligne.

Ainsi les différentes démarches présentées plus haut se différencient essentiellement par leurs approximations sur l'importance relative des termes figurant dans l'expression de la commande. Nous nous sommes intéressés aux deux premières caractéristiques énoncées plus haut et aux termes de la gravité. Pour préciser ces approximations notre grandeur de commande s'écrit :

$$\Gamma_i = A_{ii} G_i (H_i . \overset{\circ}{\varepsilon}_i + \varepsilon_i) + Q_i + \sum_{j=1}^{n} A_{ij} \overset{\circ\circ}{\theta}_{dj} \qquad \text{(III.5.4)}$$

Les tests effectués sur notre robot (M.A.23 25kg) montrent qu'un contrôle dense (fréquence d'échantillonnage $\geq$ 100 Hz) dans le cadre de cette commande, recouvre la quasi-totalité des performances (erreur dynamique inférieure à 0,5 degrés sur l'articulation la plus défavorable) attendus en utilisant l'équation globale.

III.5.2 - Découplage au niveau du vecteur de l'effecteur

Ce principe de découplage (III.5.1) peut s'appliquer au niveau de la représentation de l'effecteur E. Pour cela on exprime le vecteur des accélérations $\ddot{E}$ en fonction des vecteurs $\ddot{\theta}$ et $\dot{\theta}$.

$$\ddot{\theta} = J^{-1}\ddot{E} - J^{-1}\dot{J}\dot{\theta} \quad \text{car} \quad \dot{E} = J\dot{\theta} \; ; \; \dot{J} = \frac{d}{dt}J$$

En posant l'équation du système découplé au niveau du vecteur E nous avons :

$$\ddot{\varepsilon}_E + GH.\dot{\varepsilon}_E + G.\varepsilon = 0 \quad \text{avec} \quad \varepsilon_E = E_d(t) - E(t)$$

Nous obtenons l'expression du vecteur de commande réalisant ce découplage :

$$\Gamma = \Gamma_{NL} + AJ^{-1}\{GH.\dot{\varepsilon}_E + G\varepsilon_E + \ddot{E}_d - J\ddot{\theta}\}$$

Ce principe fut étudié par Khatib (6) qui montra son intérêt lorsque le vecteur de l'effecteur est choisi de façon judicieuse. Cependant sa mise en oeuvre et ses performances sont équivalentes à celles du découplage présenté dans le paragraphe suivant.

III.6 - APPLICATION A LA COMMANDE DYNAMIQUE D'UN MANIPULATEUR

Notre application à la commande dynamique consiste à ajouter au système de coordination, précédemment développé, le contrôle dynamique dont la grandeur de commande est donnée

par l'équation III.5.4 :

$$\Gamma_i = \underbrace{A_{ii} G_i \left(H_i \overset{\circ}{\varepsilon}_i + \varepsilon_i\right)}_{\Gamma_{i_1}} + \underbrace{Q_i + \sum_j^n A_{ij} \overset{\circ\circ}{\theta}_j}_{\Gamma_{i_2}}$$

Pour réaliser l'action corrective Γ_{i1} il est nécessaire de moduler les caractéristiques (gains et amortissements) de la régulation de position. La structure que nous avons conçue consiste à remplacer les éléments de gains classiques par des multiplieurs analogiques. Ces derniers réalisent des gains variables programmables par calculateur. Il s'agit d'une structure hybride permettant un câblage très simple du terme Γ_{i1}.

Le terme Γ_{i2} est calculé entièrement sur notre unité microprocesseur. Pour cela nous calculerons, en ligne, les vecteurs $\overset{\circ\circ}{\theta}_d$, $Q(\theta)$ et la matrice $A(\theta)$ conformément au procédé exposé dans le paragraphe (III.3).

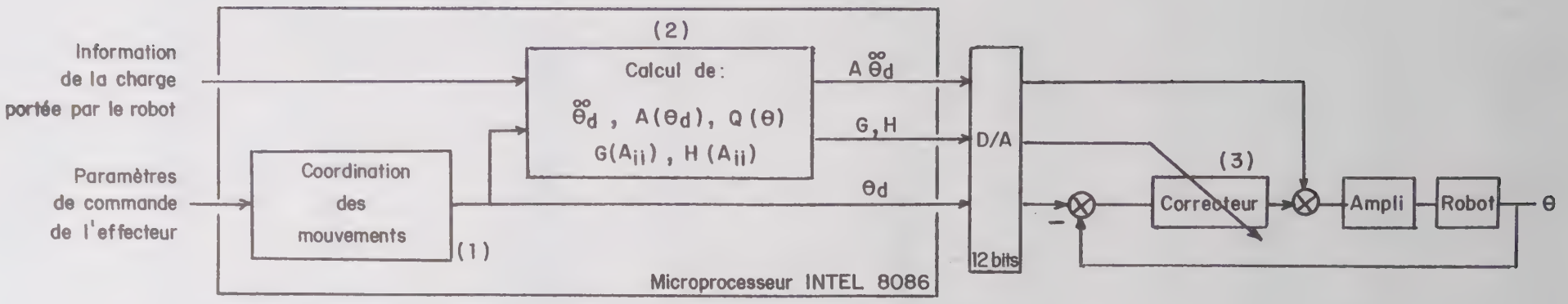
Le schéma global de notre système de commande est montré sur la figure (III.1).

L'implantation du contrôle dynamique se situe en aval du système de coordination développé aux chapitres I et II. Sa structure est compatible avec les fonctions de coordination pour lesquelles il n'introduit aucune modification structurale.

L'application est effectuée pour le manipulateur M.A.23 25 kg. Nous montrons sur les figures III.2, III.3 et III.4 respectivement :

- l'évolution des consignes relatives aux trois premiers moteurs et les réponses correspondantes du robot
- l'évolution des consignes relatives aux trois derniers moteurs et les réponses correspondantes du robot
- l'évolution des couples dynamiques, d'origine inertielle, appliqués sur les 6 organes moteurs.

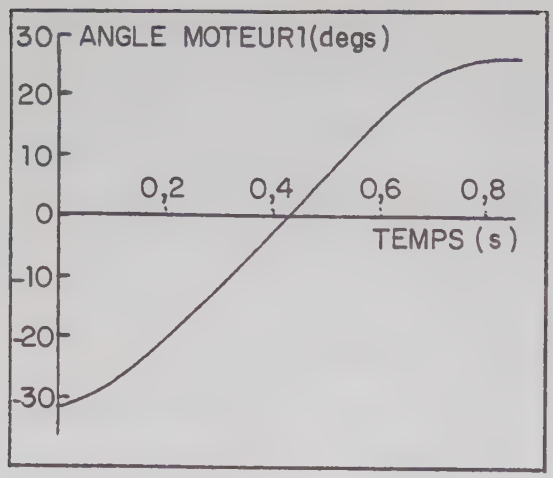
Le robot porte une charge de 8 kg.

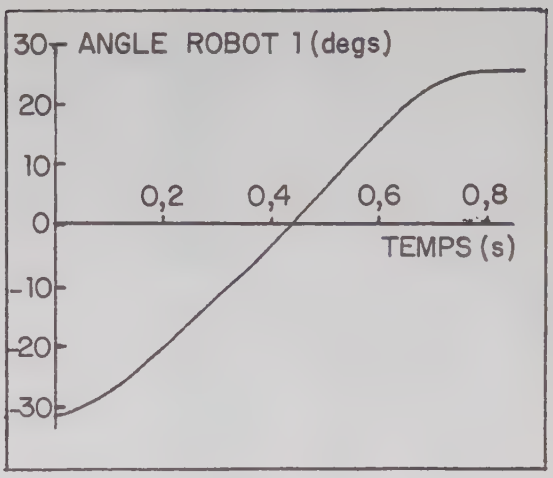


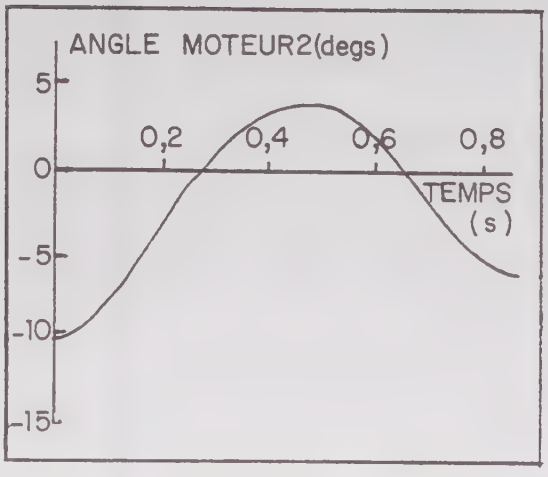
Description :

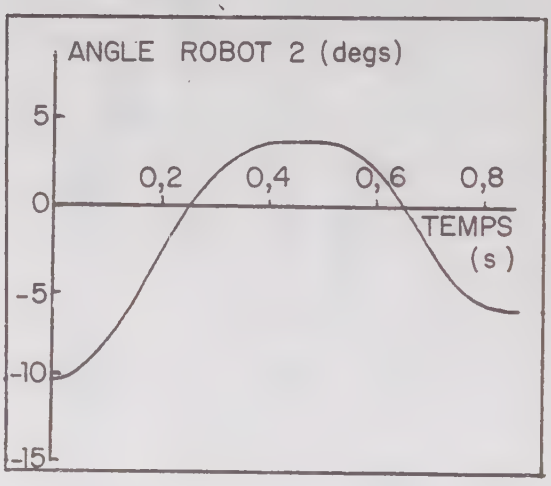
1. Coordination des mouvements : élaboration de trajectoires d'évolution, des variables généralisées θ_d, à partir d'une description d'une configuration de l'effecteur. Le cycle élémentaire dure 8 ms.
2. Calcul des signaux dynamiques : Calcul de la matrice d'inertie A(θd), du vecteur des accélérations $\ddot{\theta}_d$, du vecteur de la gravité Q(θd) et de la matrice des gains des correcteurs. Le cycle dure 9 ms.
3. Correcteurs analogiques "proportionnels et dérivés" à gains et amortissements programmables : utilise les multiplieurs analogiques dans la réalisation de gains variables.

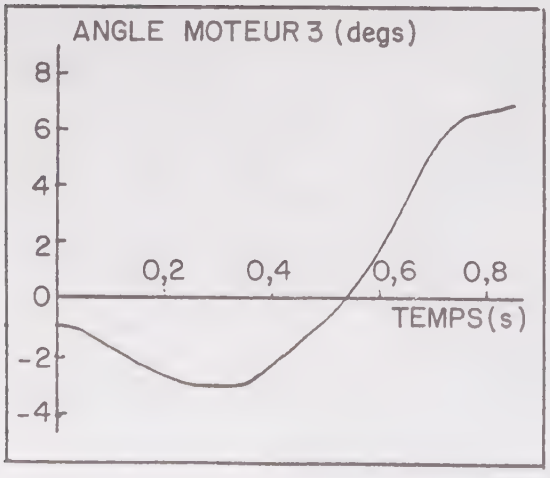
Figure III.1 : Coordination et commande dynamique par microprocesseur











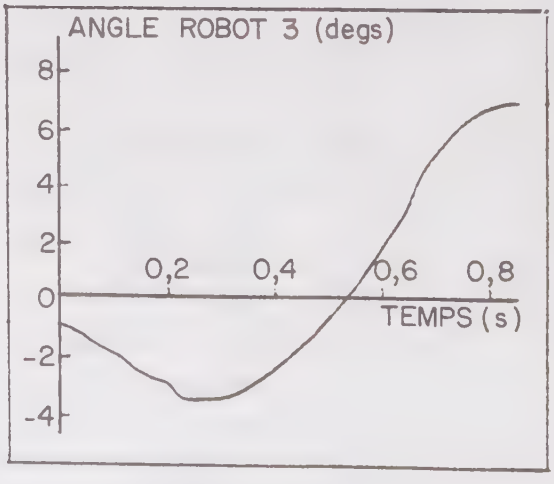
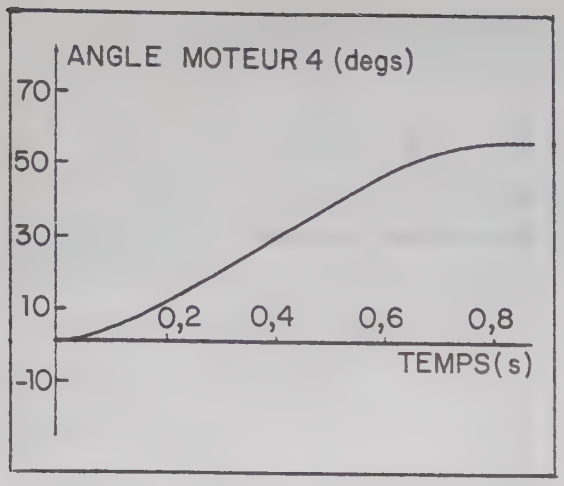
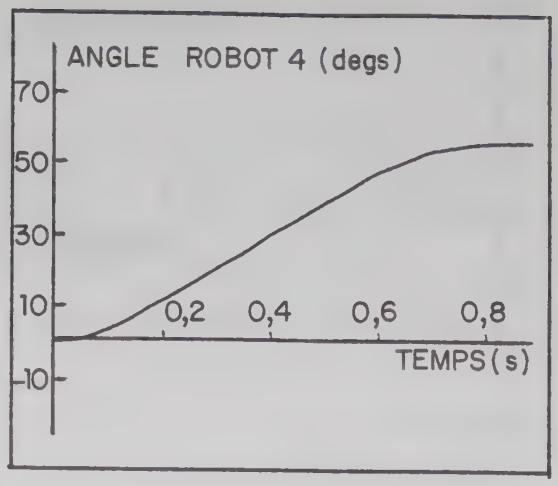
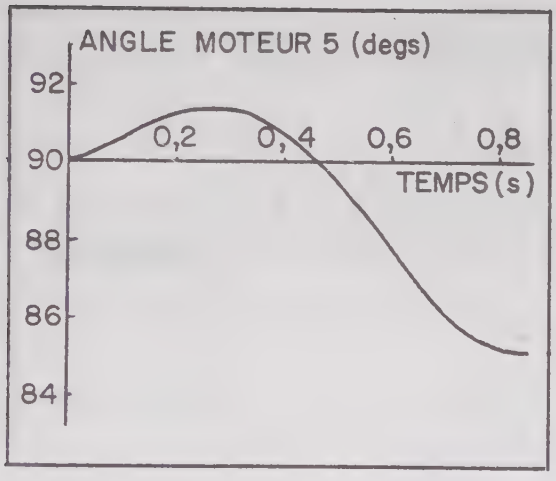
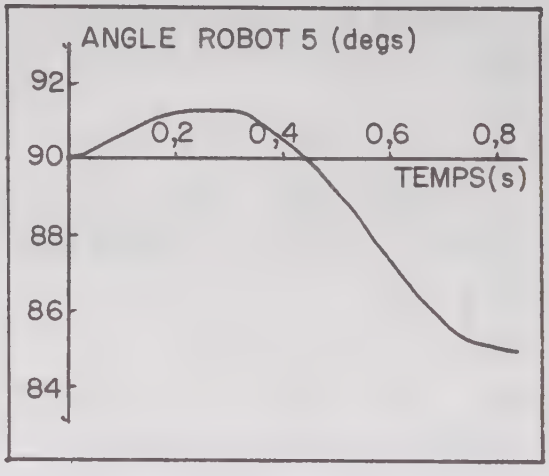


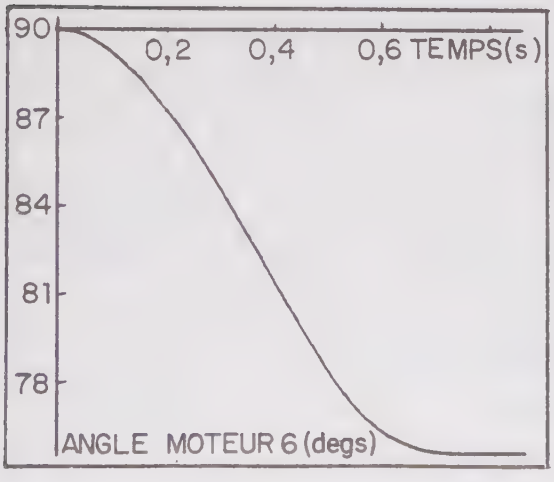
Figure III.2 : Evolution des angles de consigne des moteurs et celles du robot











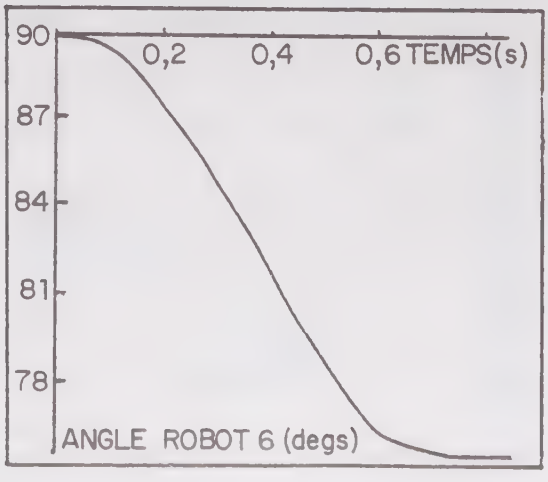
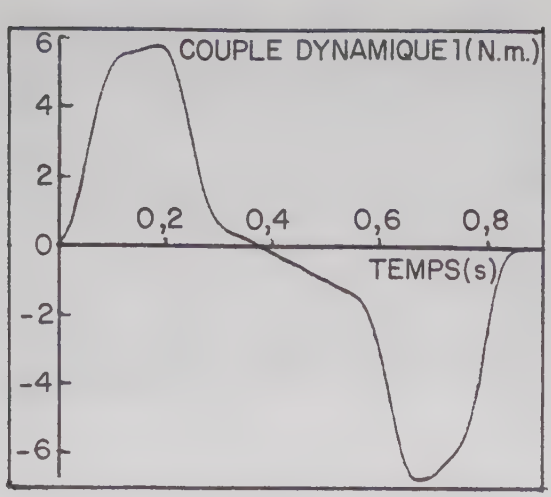
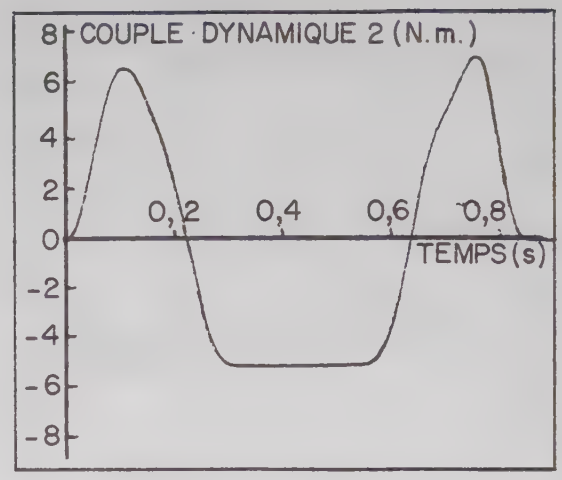
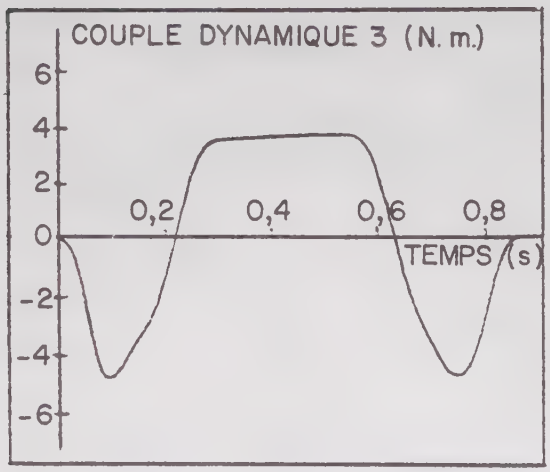
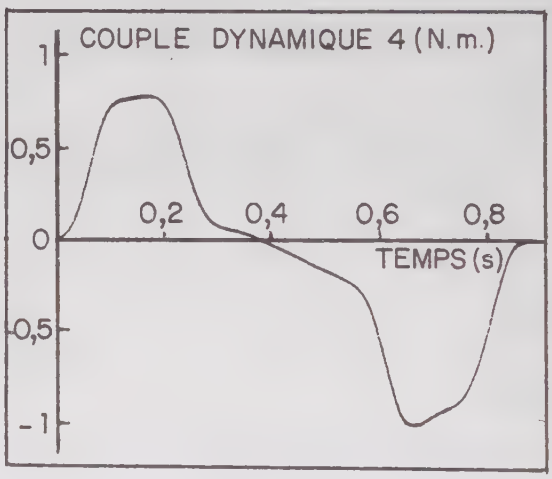


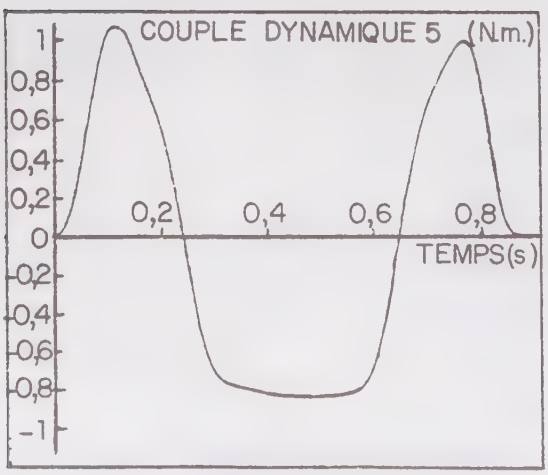
Figure III.3 : Evolution des angles de consigne des moteurs et celles du robot











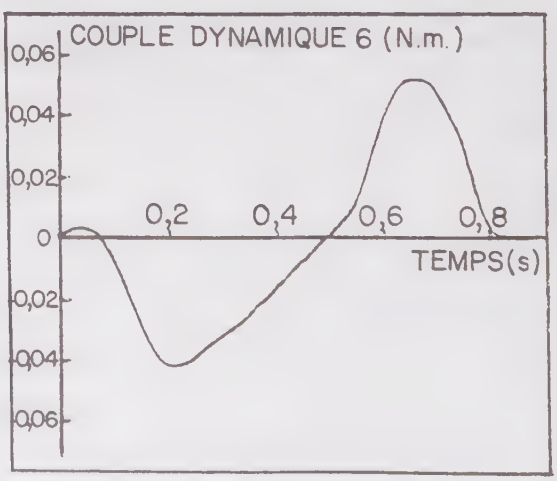


Figure III.4 : Evolution des couples dynamiques relatifs aux moteurs

III.6.1 - Amélioration des performances

La nature non linéaire du changement de coordonnées conduit au niveau des variables généralisées à des lois d'évolutions fortement différentes de celles imposées à l'évolution des paramètres de l'effecteur (ou paramètres de commande). Ainsi l'importance du terme $\sum_j A_{ij} \ddot{\Theta}_j$ n'est perceptible que si les accélérations $\ddot{\Theta}_j$ (j : 1,...,6) ou/et la charge sont suffisamment grandes. Pour mettre en valeur l'apport du découplage inertiel pour chacunes des variables généralisées nous avons choisi d'effectuer des mouvements satisfaisant les conditions suivantes :

- le robot manipulateur porte une charge parallélépipédique de 8 kg

- le mouvement est effectué aussi rapidement pour permettre de distinguer les réponses de transitoires obtenues avec et sans le découplage inertiel

- le choix de la position et de l'orientation est dicté par le souci d'obtenir un débattement suffisamment grand sur telle variable généralisée.

Nous comparerons sur les figures présentées plus loin les évolutions obtenues des angles de consignes (issues du système de coordination), des angles du robot sans le découplage inertiel, et ensuite ces derniers lorsque le découplage est appliqué.

Les figures III.5, III.6 et III.7 représentent les courbes d'évolution pour les angles moteurs 1, 2 et 3. Le mouvement commandé permet de passer de E_I à E_F qui sont les suivantes :

$$E_I = (-400,600,550,-1000,0,0)^t \qquad E_F = (400,600,550,1000,0,0)^t$$

La figure III.8 montre l'évolution de l'angle moteur 4. Le mouvement est défini par :

$E_I = (0,600,550,1000,0,0)^t$ $E_F = (0,600,550,-1000,0,0)^t$

La figure III.9 montre l'évolution de l'angle moteur 5. Le mouvement est défini par :

$E_I = (0,600,550,0,600,0)^t$ $E_F = (0,600,550,0,-700,0)^t$

La figure III.10 montre l'évolution de l'angle moteur 6. Le mouvement est défini par :

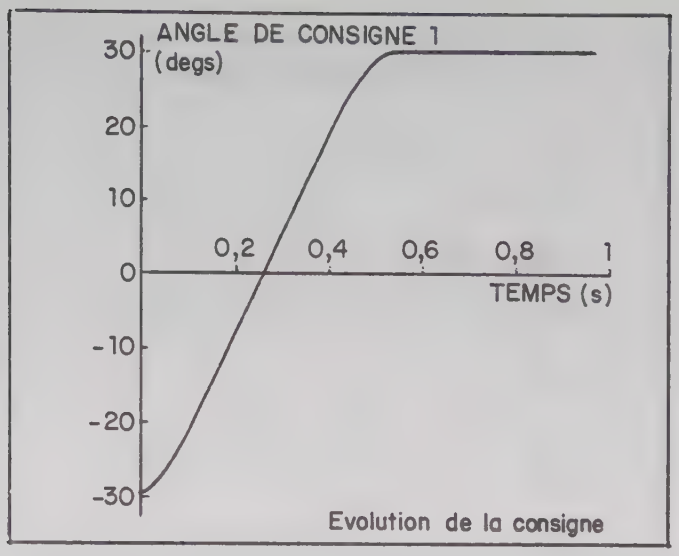
$E_I = (0,600,550,0,0,1000)^t$ $E_F = (0,600,550,0,0,-1000)^t$

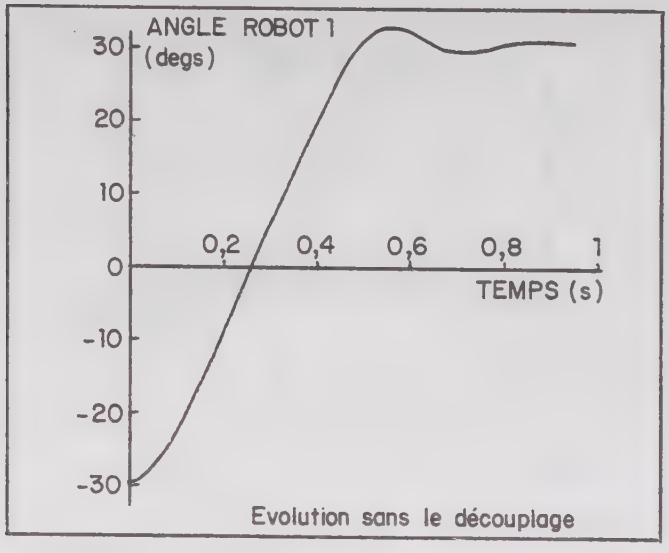
En comparant les réponses obtenues, avec et sans le découplage inertiel, nous observons deux remarques qualitatives :

- une meilleure tenue des transitoires notamment lorsque l'évolution locale de ces angles correspond à un polynome de degrés supérieur à 1

- le freinage, dans la phase finale du transitoire, se fait pratiquement sans dépassement et l'angle réel converge plus vite vers la position de consigne.

Enfin les figures III.11, 12 et 13 montrent une comparaison entre les angles de consignes, les angles moteurs et les angles d'articulations (potentiomètres placés sur les articulations).


ANGLE DE CONSIGNE 1
(degs)
30
20
10
0
-10
-20
-30
0,2
0,4
0,6
0,8
1
TEMPS (s)
Evolution de la consigne


ANGLE ROBOT 1
(degs)
30
20
10
0
-10
-20
-30
0,2
0,4
0,6
0,8
1
TEMPS (s)
Evolution sans le découplage

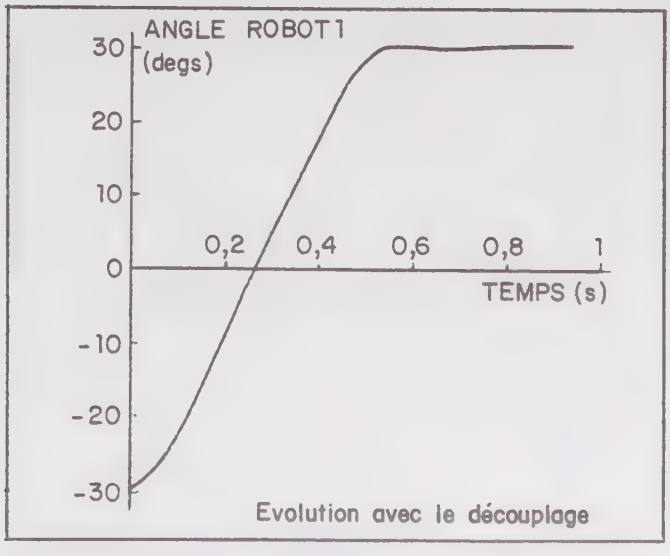
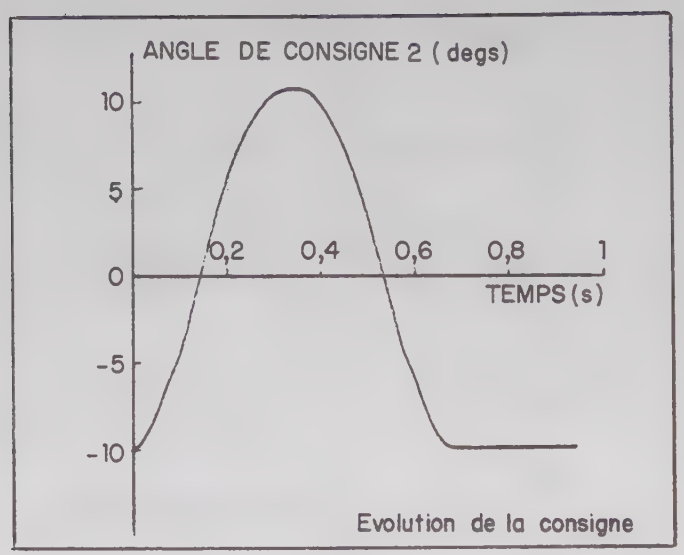
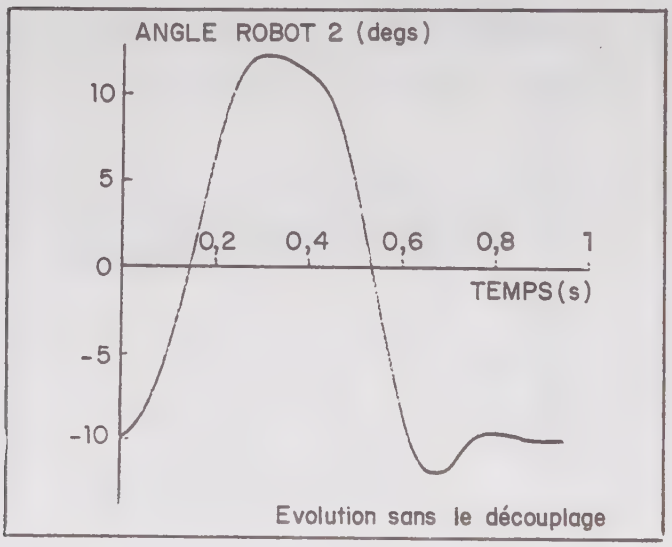

ANGLE ROBOT 1
(degs)
30
20
10
0
-10
-20
-30
0,2
0,4
0,6
0,8
1
TEMPS (s)
Evolution avec le découplage

Figure III_5


ANGLE DE CONSIGNE 2 (degs)
10
5
0
-5
-10
0,2
0,4
0,6
0,8
1
TEMPS (s)
Evolution de la consigne


ANGLE ROBOT 2 (degs)
10
5
0
-5
-10
0,2
0,4
0,6
0,8
1
TEMPS (s)
Evolution sans le découplage

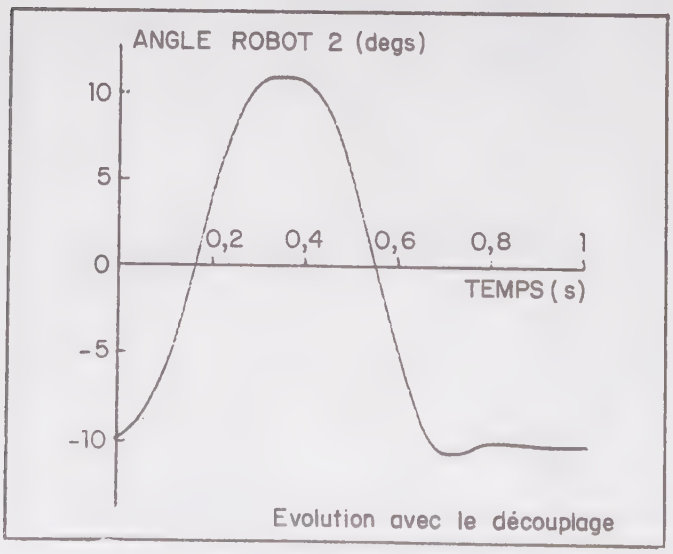
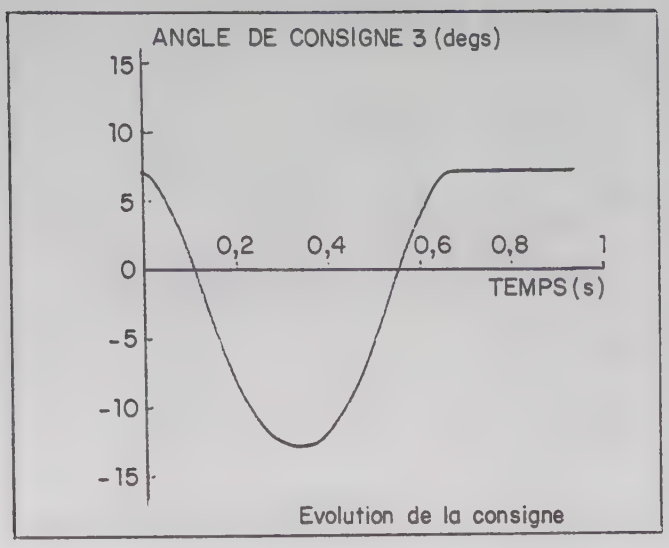
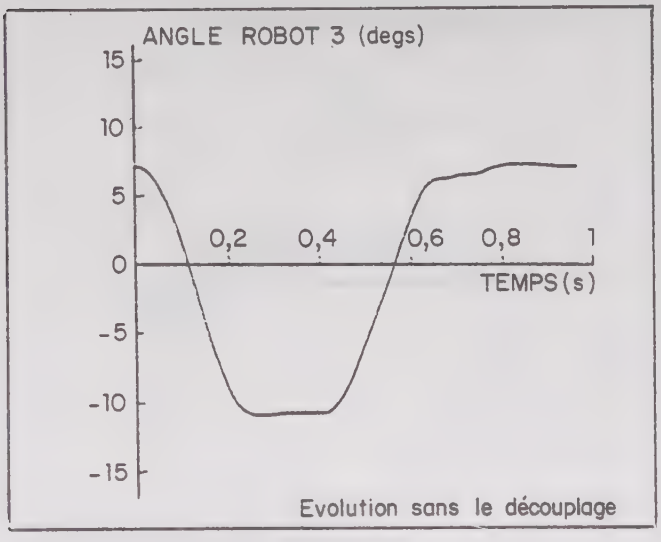

ANGLE ROBOT 2 (degs)
10
5
0
-5
-10
0,2
0,4
0,6
0,8
1
TEMPS (s)
Evolution avec le découplage

Figure III.6


ANGLE DE CONSIGNE 3 (degs)
15
10
5
0
-5
-10
-15
0,2
0,4
0,6
0,8
1
TEMPS (s)
Evolution de la consigne


ANGLE ROBOT 3 (degs)
15
10
5
0
-5
-10
-15
0,2
0,4
0,6
0,8
1
TEMPS (s)
Evolution sans le découplage

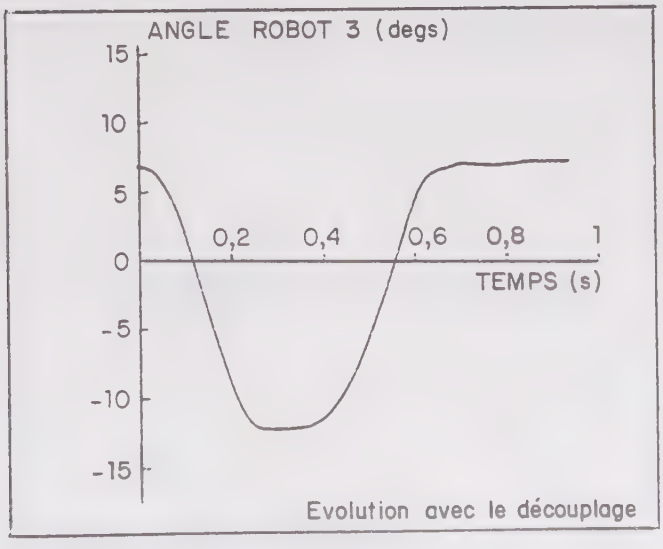
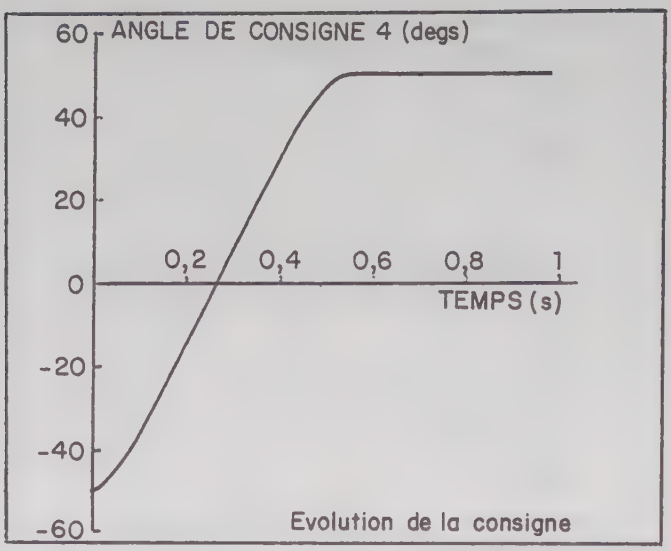
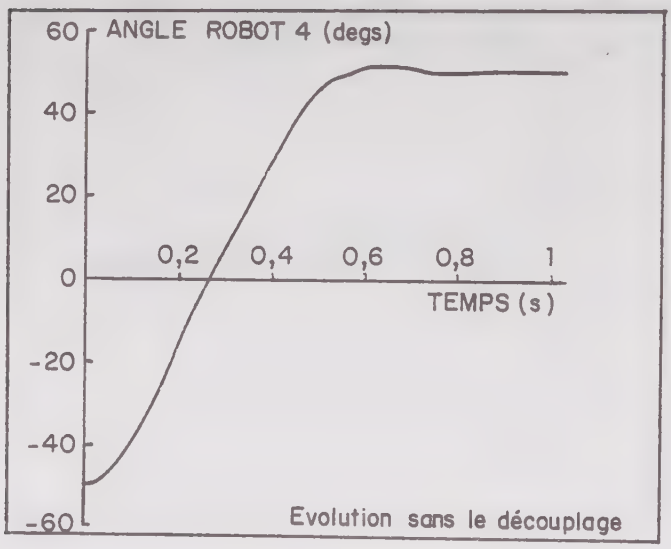

ANGLE ROBOT 3 (degs)
15
10
5
0
-5
-10
-15
0,2
0,4
0,6
0,8
1
TEMPS (s)
Evolution avec le découplage

Figure III.7


60
ANGLE DE CONSIGNE 4 (degs)
40
20
0
-20
-40
-60
0,2
0,4
0,6
0,8
1
TEMPS (s)
Evolution de la consigne


60
ANGLE ROBOT 4 (degs)
40
20
0
-20
-40
-60
0,2
0,4
0,6
0,8
1
TEMPS (s)
Evolution sans le découplage

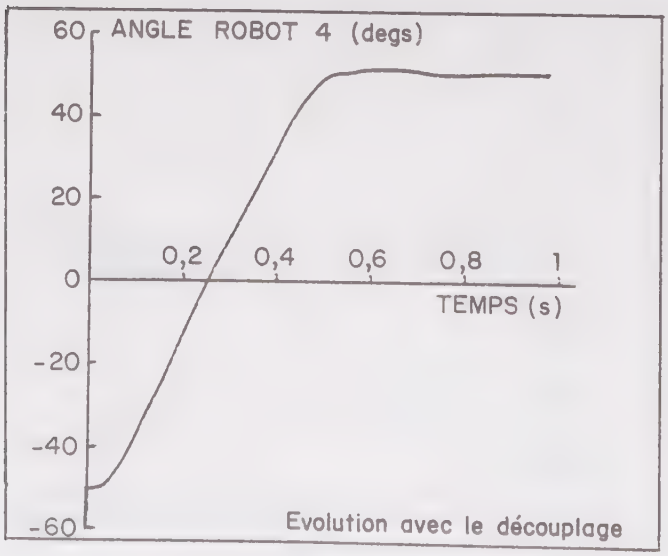
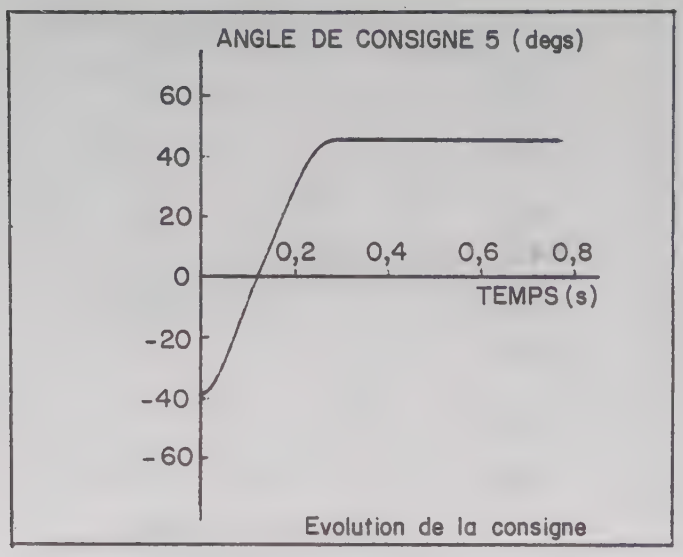
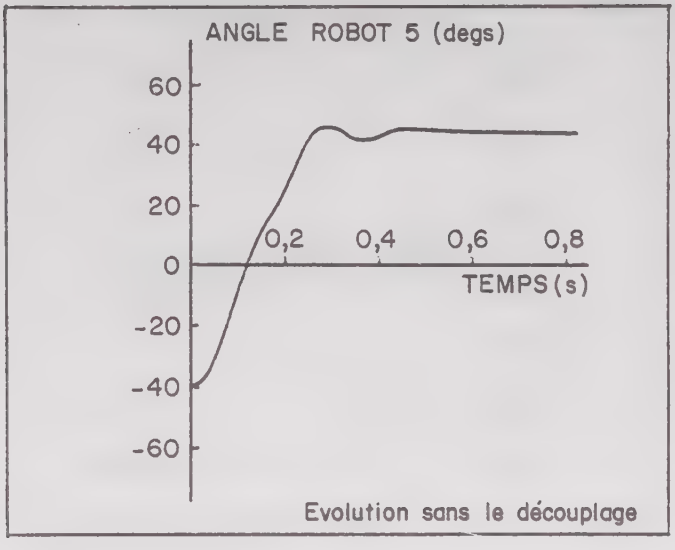

60
ANGLE ROBOT 4 (degs)
40
20
0
-20
-40
-60
0,2
0,4
0,6
0,8
1
TEMPS (s)
Evolution avec le découplage

Figure III . 8


ANGLE DE CONSIGNE 5 (degs)
60
40
20
0
-20
-40
-60
0,2
0,4
0,6
0,8
TEMPS (s)
Evolution de la consigne


ANGLE ROBOT 5 (degs)
60
40
20
0
-20
-40
-60
0,2
0,4
0,6
0,8
TEMPS (s)
Evolution sans le découplage

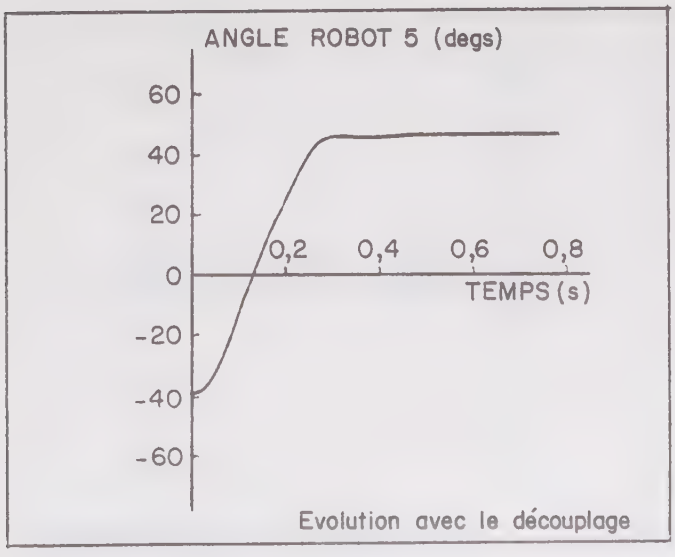
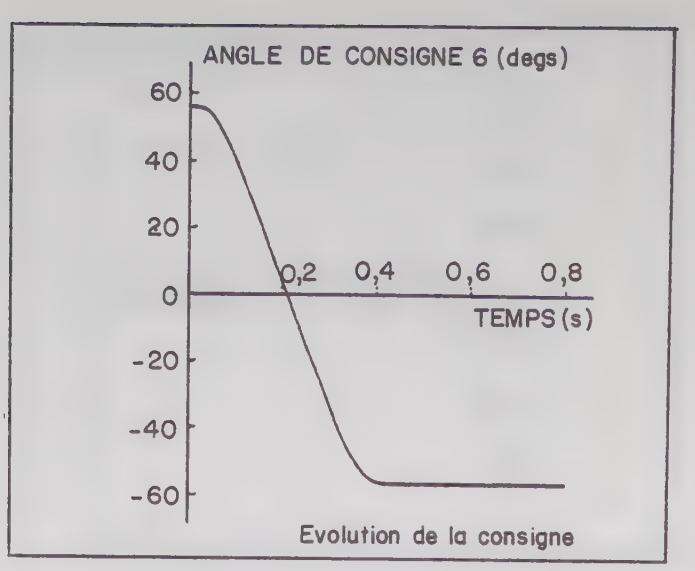
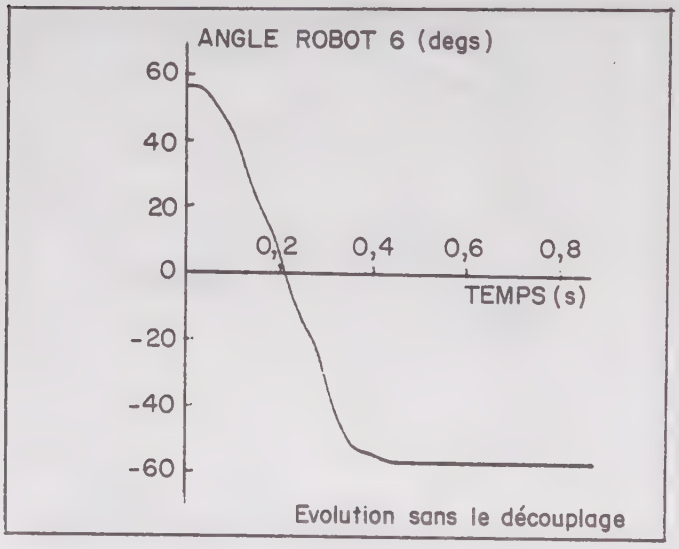

ANGLE ROBOT 5 (degs)
60
40
20
0
-20
-40
-60
0,2
0,4
0,6
0,8
TEMPS (s)
Evolution avec le découplage

Figure III. 9





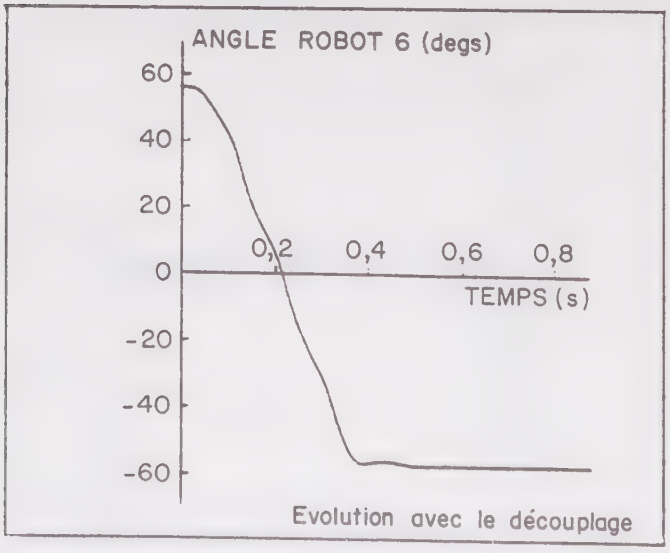
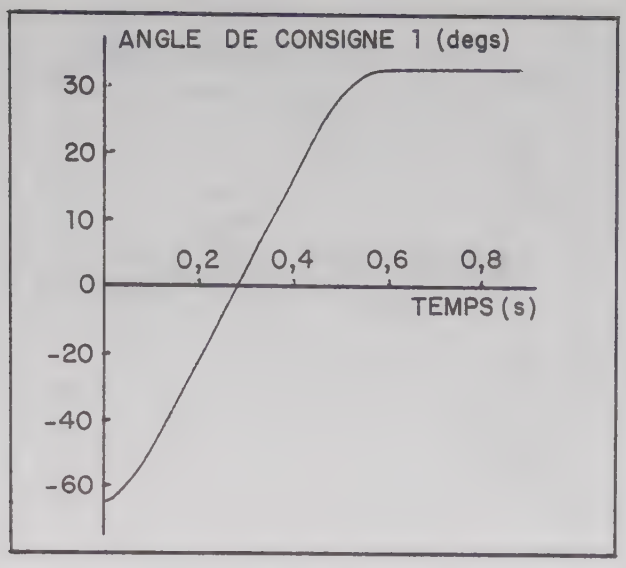
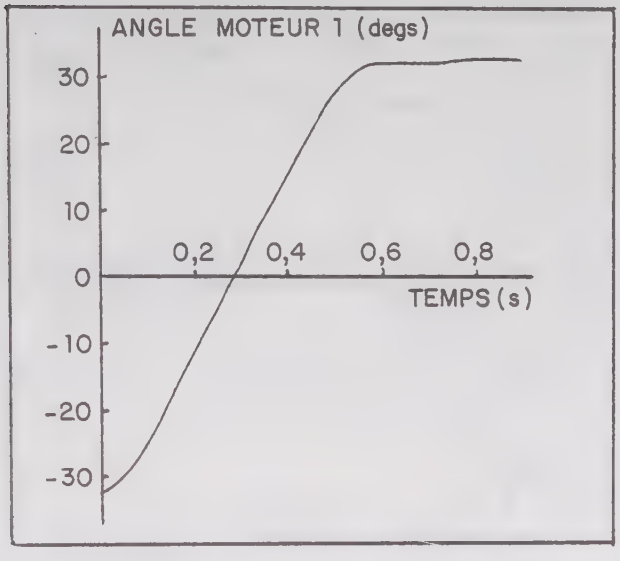


Figure III. 10





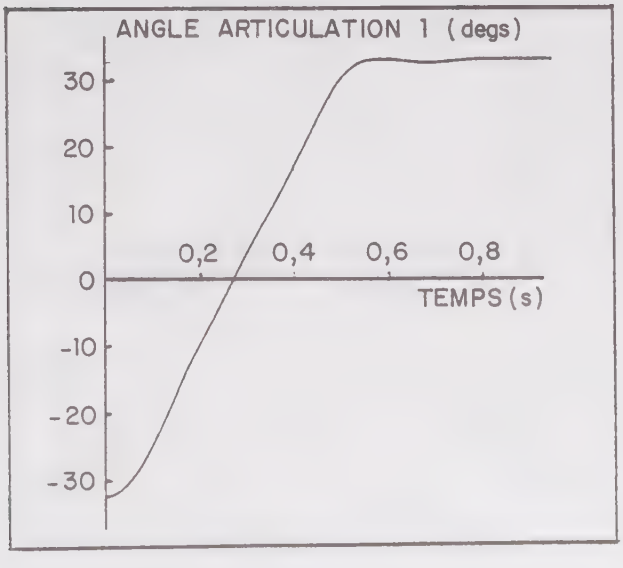


Figure III_11

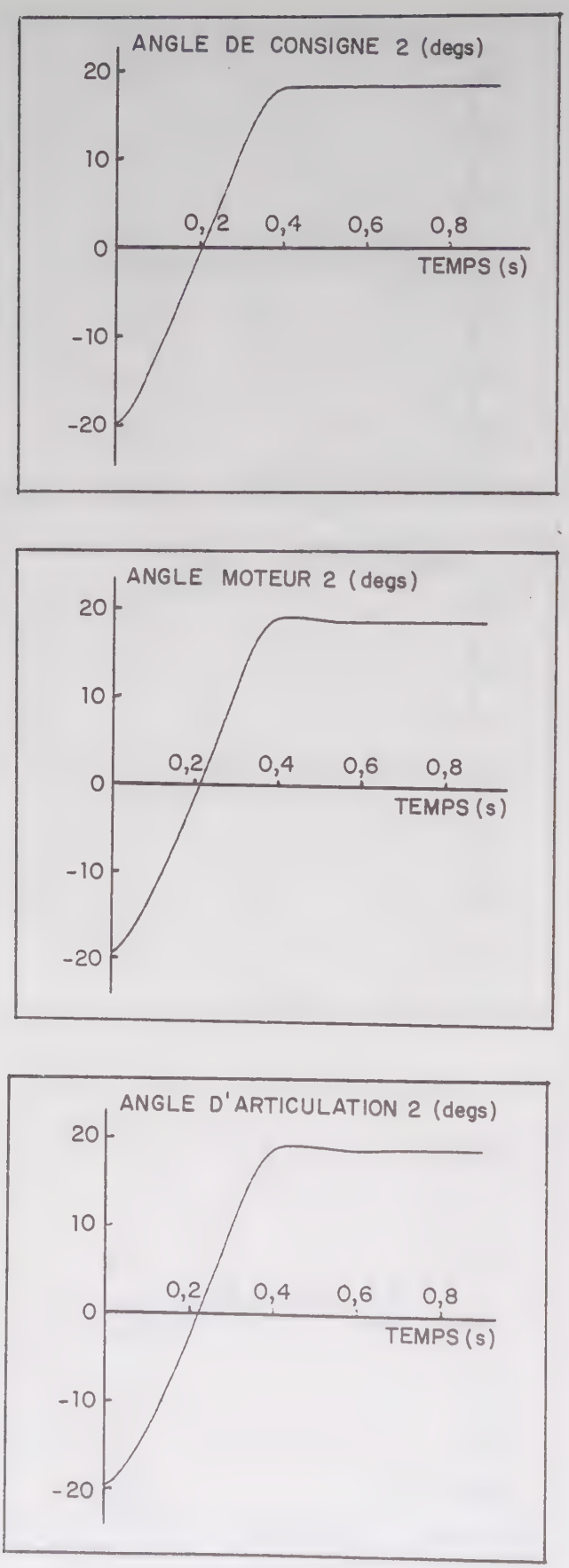
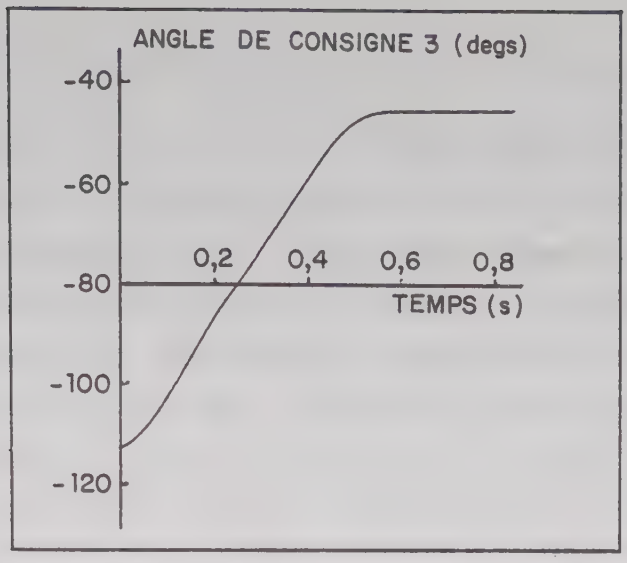
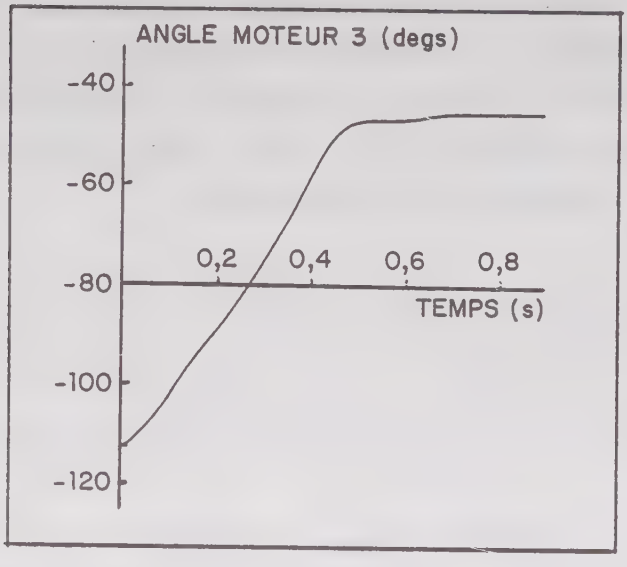


Figure III. 12





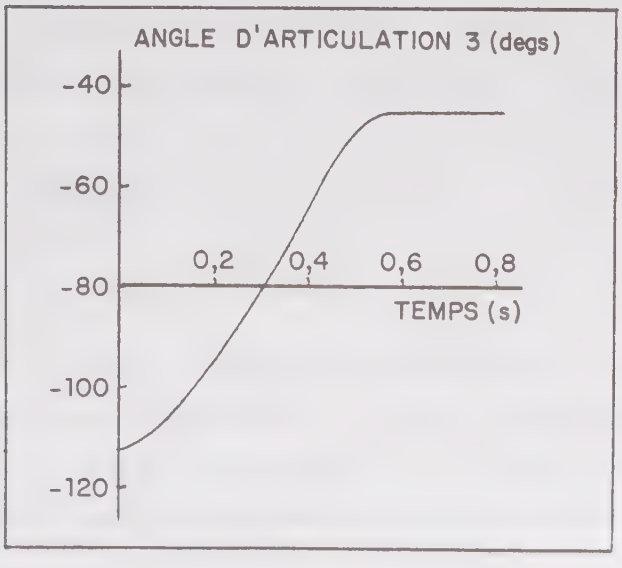


Figure III . 13

III.7 - CONCLUSION

Nous avons étudié dans ce chapitre les problèmes liés à l'établissement d'un contrôle dynamique pour un système articulé. Pour cela nous avons analysé, de façon générale, l'obtention de l'équation de mouvement à partir d'un modèle mathématique utilisant le formalisme de Lagrange. Ce dernier permet l'écriture, sous une forme littérale, des forces dynamiques dont le système articulé est le siège quand il est en mouvement. Nous avons ensuite montré la complexité de ces expressions et la nécessité de méthodes plus directes. Ainsi nous avons proposé d'exploiter le système articulé pour mesurer certaines forces d'interaction et d'en établir un modèle de comportement simple. Notre approche conduit, lors d'une période d'apprentissage, à l'établissement d'un modèle discret pour les paramètres qui interviennent dans l'équation de mouvement.

Notre méthode a le mérite de s'intéresser davantage aux effets réels qui apparaissent dans le comportement dynamique du système. Elle permet en outre de suivre l'évolution de ces forces lors du vieillissement du matériau mécanique.

Sur le plan fonctionnel nous avons montré, dans le cadre de notre application, la possibilité d'un découplage inertiel pour un manipulateur à six degrés de liberté. Pour cela nous avons implanté notre algorithme de contrôle sur une unité à microprocesseur performant. Cette dernière élabore, en temps réel, les signaux nécessaires pour compenser les forces de couplage et maintenir les performances constantes au cours du mouvement.

Les courbes d'évolution exposées montrent la fidélité entre l'évolution désirée, issue du système de coordination, et l'évolution réelle du robot. Ces dernières montrent un meilleur suivi de trajectoires que celui obtenu avec les régulations traditionnelles.

Avec notre approche réaliste, et la commande en temps

réel que nous avons réalisée nous pensons qu'une amélioration sensible est obtenue sur le comportement dynamique et statique des robots manipulateurs.

CONCLUSION GENERALE

Nos travaux ont eu pour objectif l'étude du comportement adaptatif des robots manipulateurs et l'accroissement de leurs performances statiques et dynamiques.

L'analyse des éléments constituants dans les fonctions adaptatives nous a permis d'approfondir les notions de coordination des mouvements, d'analyse d'effort, de conception de capteurs ainsi que de la recherche d'un contrôle automatique plus performant.

Dans la première partie de cette thèse nous avons exposé l'établissement des modèles géométriques et variationnels des robots manipulateurs. Nous avons ensuite analysé les problèmes liés à l'inversion de ces formes de représentation qui conduisent à l'élaboration du changement de coordonnées. Autour de cette dernière étape de la commande nous avons proposé une structure de coordination comprenant des fonctions de mouvements globaux et d'autres adaptées aux mécanismes rebouclés. Dans notre approche le centre d'action peut être modifié aisément ce qui permet de réaliser des mouvements complexes avec une commande simple. La généralisation de la notion de centre d'action constitue un pas intéressant dans la mise en oeuvre des fonctions d'assemblage automatique. Elle permet en outre de produire des mouvements complexes qui sont souvent à la charge de l'utilisateur dans les systèmes traditionnels. Nous avons montré ensuite comment une telle structure peut être implantée sur un matériel de très bas niveau tel que les microprocesseurs. Les performances obtenues, dans le cadre de nos applications, sont tout-à-fait compétitives tant sur le plan de la précision des trajectoires que sur le plan de la rapidité du cycle élémentaire de coordination (cycle de commande en 8 ms).

Dans la deuxième partie nous avons étudié le principe d'une régulation en effort. Nous avons dégagé les moyens de calcul de l'information d'effort, au niveau de l'organe de préhension, en suivant deux méthodologies : utilisation des grandeurs intrinsèques au robot (les couples moteurs) et utilisation d'un capteur de forces. Nous avons conçu ce dernier dans le but de déduire les forces extérieures qui agissent sur l'objet manipulé.

L'analyse de l'asservissement en effort dans le cas des opérations d'assemblage nous a permis de dégager les mécanismes élémentaires de régulation qui sont indispensables pour accomplir les phases d'affinement dans ces opérations. Nous avons ensuite proposé une structure pour ces mécanismes, qui permet de combiner une commande en position selon certains axes du mouvement avec une commande en effort selon d'autres axes. Les corrections de position et d'orientation peuvent s'effectuer aisément avec notre méthode où le centre d'action se confond avec le centre de compliance. En plaçant le centre d'action au point de contact des pièces à assembler les corrections citées plus haut permettent alors, et de façon unique, de réduire ces forces de contacts.

Nous avons présenté des applications aux fonctions adaptatives telles que l'insertion, l'extraction et le poser.

Notre approche a l'avantage d'être inspirée par les besoins réels rencontrés dans le problème de l'assemblage automatique. Le capteur de forces et le logiciel de régulation que nous proposons présentent un apport intéressant, à notre avis, pour l'obtention d'une compliance active en robotique.

Dans la troisième partie nous avons étudié l'équation de comportement d'un système articulé. Cette dernière met en évidence l'existence de forces de couplages diverses. Celles-ci réduisent la précision dynamique et statique du robot.

Les méthodes de commande par le calcul du modèle (modèle de paramètres) conduisent à une représentation complexe des paramètres intervenant dans l'équation de comportement. Cela rend difficile la commande en temps réel. De plus se pose le problème de la fidélité du modèle étant donné la complexité de la machine électro-mécanique.

Afin de surmonter ces difficultés nous avons proposé une méthode basée sur l'identification des paramètres de l'équation de comportement lors d'une phase d'apprentissage effectuée sur le robot. Après avoir dégagé un modèle simple pour décrire l'interaction entre l'organe moteur et l'articulation nous avons établi l'ordonnancement selon lequel ces paramètres sont estimés pour une configuration donnée du système articulé. Cette méthode permet notamment d'identifier les paramètres d'inertie tels qu'ils sont ressentis par les organes moteurs. Elle s'intéresse donc aux effets réels qui affectent le système pendant son mouvement.

L'apprentissage des paramètres conduit à une représentation discrète de ces derniers. La technique d'interpolation multivariable nous a fourni un moyen pour obtenir ces paramètres de façon continue, pour l'élaboration de la commande.

Le contrôle dynamique résulte d'un objectif de découplage ou de compensation des forces dont le robot est le siège. Ce principe, qui est généralement appliqué à l'équation temporelle, fournit une expression simple de la commande. Pour mettre en oeuvre cette dernière nous avons proposé une structure d'asservissement adaptée à la commande par calculateur des systèmes articulés. Elle permet en particulier de programmer les caractéristiques (gains et amortissements), de la boucle de régulation, en fonction des charges.

Nos applications, à ce sujet, ont porté sur le contrôle dynamique d'un robot à six degrés de liberté sur une unité à microprocesseur. Dans son schéma d'implantation cette commande se situe en aval de notre système de coordination qui

engendre les trajectoires à suivre. Cette unité élabore les divers signaux pour réaliser le découplage et maintenir les performances constantes au cours du mouvement.

Les résultats théoriques et pratiques présentés dans ce mémoire donnent une idée incomplète du travail réel que nous avons effectué. En particulier nous avons eu à développer la totalité du système matériel tel que les capteurs d'effort et de proximité, les cartes d'asservissement adaptées à la commande dynamique, les interfaces et la bibliothèque mathématique.

Par ces composantes théoriques et pratiques nous pensons que nous avons atteint l'objectif qui nous a été fixé. Nous espérons que les structures que nous avons analysées et synthétisées représentent une contribution intéressante et positive au problème général de la commande automatique en robotique.

BIBLIOGRAPHIE

(1) W. KHALIL
"Modélisation et commande par calculateur du manipulateur M.A.23 - Extension à la conception par ordinateur des manipulateurs"
Thèse de Doc. Ingénieur, U.S.T.L., Montpellier, sept. 1976

(2) W. KHALIL
"Contribution à la commande automatique des manipulateurs avec l'aide d'un modèle mathématique des mécanismes"
Thèse d'Etat, U.S.T.L., Montpellier, oct. 1978

(3) A. FOURNIER
"Génération de mouvements en robotique . Applications des inverses généralisées et des pseudo-inverses"
Thèse d'Etat, U.S.T.L., Montpellier, 1980

(4) P. COIFFET
"Les robots"
Tome 1 et 2, Hermes Publishing (France), 1981

(5) A. LIEGEOIS, M.J. ALDON
"Génération et programmation automatiques des équations de Lagrange des robots et manipulateurs"
Rapport IRIA

(6) O. KHATIB
"Commande dynamique dans l'espace opérationnel des robots manipulateurs en présence d'obstacles"
Thèse Doc. Ingénieur, E.N.S.A.E., Déc. 1980

(7) D. FALEK, M. PARENT
"LAMA-S : an evolutive language for an intelligent robot"

Sém. Inter. sur les méthodes et langages de programmation des robots industriels, Juin 1979, IRIA, Rocquencourt, France

(8) D.L. PIEPER
"The kinematics of manipulators under computer control"
Ph.D. Thesis, Stanford Univ., 1968

(9) B. ROTH
"Performance evaluation of manipulators from a Kinematic view point"
Cours de robotique, IRIA, Toulouse, sept. 1976

(10) B. SHIMANO, B. ROTH
"Ranges of motion of manipulators"
2d CISM IFT.MM Symp. on Theory and practice of robots and manipulators, Warsaw, Sept. 1976

(11) M. RENAUD
"Contribution à la modélisation et à la commande des robots manipulateurs"
Thèse d'Etat, Toulouse, Sept. 1980

(12) C.J.M. WILLIAMS
"Dynamics modelling and formulation techniques for non-rigid spacecraf"
E.S.A. Symp. on the dynamics and control on non-rigid space vehicles, Frascati, Italy, mai 1976

(13) M.B. IGNAT'YEV
"Robot-manipulator control algorithms"
Joint pub. Research service, Virginia, Aug. 1973

(14) D.E. WHITNEY
"Resolved motion rate control of manipulators and human prostheses"
I.E.E.E. Trans. on Man Machines Systems, 10, vol. 2, juin 1969

(15) YU.M. KOZLOV, A.P. MAYROV
"A system for programmed control of an extremity of a robot"
Engineering cybernetics, n° 5, 1976

(16) A. LIEGEOIS
"Commande des systèmes mécaniques articulés"
RAIRO Série Automatique, Fév. 1975

(17) T.N.E. GREVILLE
"Some applications of the pseudo-inverse of a matrix"
SIAM Revue, II 1960

(18) A. LIEGEOIS, M. RENAUD
"Synthèse de la commande dynamique d'un téléopérateur redondant"
Revue RAIRO, J-2, mai 1975

(19) R.P. PAUL
"The theory and practice of robot manipulator programming and control"
18th IEEE conf. on Decision and Control, Fort Lauderdale, Déc. 1979

(20) J. YCLON
"Système d'expérimentation SPARTACUS"
Rapport IRIA N° 78-016, Mai 1979

(21) M. ALMOUHAMED
"Commande rapide pour le manipulateur M.A.23"
Le nouvel Automatisme, Janvier 1982

(22) M. ALMOUHAMED
"Coordination et commande dynamique par microprocesseurs"
12e Congrès Int. sur les Robots Industriels
Paris, 9-11 juin 1982

(23) J.L. NEVINS, D.E. WHITNEY
Exploratory research in industrial modular assembly
R.N° 741. The Charles Stark Draper laboratory, 1974

(24) J.L. NEVINS, D.E. WHITNEY
The force vector assembler concept
Int. Conf. on Robots and Manip. Syst. Undine, Sept. 1973

(25) J.L. NEVINS, D.E. WHITNEY
A scientific approach to the design of computer controlled manipulators
R.N° 837, The Charles Stark Draper laboratory

(26) S.H. DRAKE, C.W. PAUL, N.S. SERGIO
High speed robot assembly of precision parts using compliance instead of sensory feedback
Proc. of 7th Int. Symp. on Industrial Robots, oct. 1977, Tokyo

(27) T. GOTO
Precise insert operation by tactile controlled robot 'HI-T-HAND Expert 2'
Proc. of 4th Int. Symp. on Industrial robots
Nov. 1974, Tokyo

(28) P. COIFFET
Etude des relations capteurs-mouvements pour le manipulateur M.A.23
Rapport d'activité annuelle 1977, Contrat D.G.R.S.T./L.A.M

(29) P. COIFFET, A. LIEGEOIS, A. FOURNIER
Computer aided control of force-reflexive manipulators
7th IFAC Symp. on Auto. Control in space, Rottach-Egern 1976

(30) P. COIFFET, A. LIEGEOIS, E. DOMBE, P. RIVES
Manipulateurs flexibles et prise en compte de l'environnement. Comptes-rendus des journées d'Automatique du 30 au 31 janvier 1979
P.I. N° 117 I.R.I.S.A

(31) T.O. BINFORD, J.M. TENENBAUM
Computer vision - Computer, May 1973

(32) M. YACHIDA, S. TSUJI
A versatile machine vision system for complex industrial parts
IEEE Trans. on computers, sept. 1977

(33) S.H. DRAKE, R.M. SPENCER, S.N. SIMUNOVIC
Using compliance in assembly
The computer and automated systems association of S.M.E., 1979

(34) A.K. BEJCZY
Effect of hand-based sensors on manipulator control performance
2th Conf. on remotely manned systems. June 9-11, 1975

(35) R. BOLLES, R. PAUL
The use of sensory feedback in a programmable assembly system
MEMO-AIM-220, Stanford Artificial Intelligence laboratory, Oct. 1973

(36) D.E. WHITNEY
The mathematics of coordination and purposeful motions
Journal of Cybernetics and Inf. Sciences, T1, N°1, 1976

(37) M. ALMOUHAMED
Etudes d'heuristiques pour le contrôle-commande de robots
Conférence de l'AFCET sur l'intelligence artificielle et la reconnaissance des formes
Nancy, 16-19 sept.1981

(38) W.W. HOOKER, G. MARGULIES
"The dynamics attitude equations for a n-body satellite"
The Journal of the Astronautical Sciences, V.12, N°4, 1965

(39) J.J. UICKER
"Dynamic behavior of spatial linkages"
Trans. of the ASME, N° 68, Mech. 5, 1968

(40) J.C. SAMIN
"Dynamique des corps déformables en rotation"
Thèse d'Etat, Lauvain La Neuve, 1974

(41) E.P. POPOV
"Synthèse de la commande des robots utilisant les modèles dynamiques des systèmes mécaniques"
VI IFAC Symp. on Control in Space, Aug. 1974

(42) P. COIFFET, A. LIEGEOIS, E. DOMBE, P. RIVES
"Manipulateurs flexibles et prise en compte de l'environnement"
Comptes rendus des Journées d'Automatique du 30 et 31 janvier 1979
Rapport IRISA N° 117

(43) M.H. RAIBERT et B.K. HORN
"Manipulator control using the configuration space method"
The Industrial Robot, June 1978

(44) R. PAUL
"Modelling, trajectory calculation and servoing of a computer controled arm"
Stanford Intelligence Laboratory
Memo. AIM 177, Nov. 1972

(45) A.K. BEJCZY
"Robot arm dynamic and control"
Jet Propulsion Lab. , NASA Tech. Memo. 33-669, 1974

(46) E. FREUND
"Path Control for a redundant type of industrial robot"
Proc. of 7th Int. Symp. on Ind. Robots, Tokyo, 1977

(47) M.E. KAHN
"The near minimum time control of open loop articulated Kinematic chains"
Stanford Univ. Ph. D. thesis, Eng. Mech. 1970.

Manuscrit reçu le 22 février 1982

ANNEXE 1

MODELE GEOMETRIQUE INVERSE POUR LE M.A.23

Données structurales

domaines accessibles

Θ_1 [-60°, +60]

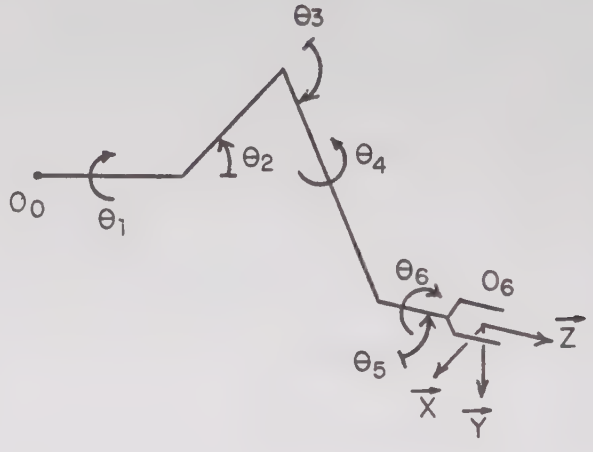
Θ_2 [-70°, 57°]

Θ_3 [-135°,179°]

Θ_4 [-179°, +179°]

Θ_5 [-29°, +109°]

Θ_6 [-180°, +180°]



Données géométriques

Vecteurs $\vec{X}$, $\vec{Y}$, $\vec{Z}$ et $\overrightarrow{O_0O_6}$:

X_X = C1C4C6-S1C23S4C6-C1S4C5S6-S1C23C4C5S6+S1S23S5S6
X_Y = S1C4C6+C1C23S4C6-S1S4C5S6+C1C23C4C5S6-C1S23S5S6
X_Z = S1S4S5-C1C23C4S5-C1S23C5

Y_X = -C1C4S6+S1C23S4S6-C1S4C5C6-S1C23C4C5C6+S1S23S5C6
Y_Y = -S1C4S6-C1C23S4S6-S1S4C5C6+C1C23C4C5C6-C1S23S5C6
Y_Z = -S23S4S6+S23C4C5C6+C23S5C6

Z_X = C1S4S5+S1C23C4S5+S1S23C5
Z_Y = S1S4S5-C1C23C4S5-C1S23C5
Z_Z = -S23C4S5+C23C5

X_6 = S1(S2L2+S23L3) + Z_X.L6
Y_6 = -C1(S2L2+S23L3) + Z_Y.L6
Z_6 = L1+C2L2+C23L3+Z_Z.L6

Calcul des variables géométriques

$$X5 = X6 - Z_X L6$$
$$Y5 = Y6 - Z_Y L6$$
$$Z5 = Z6 - Z_Z L6$$

$$TANG(\theta 1) = -X5/Y5$$
$$COS(\theta 3) = (X_5^2 + Y_5^2 + (Z5-L1)^2 - L_2^2 - L_3^2) / 2\,L2\,L3$$
$$SIN(\theta 3) = -\sqrt{1 - \cos^2(\theta 3)}$$

$$SIN(\theta 2) = -((Z5-L1)S3L3 + (L2+C3L3)Y5/C1) / (L_2^2+L_3^2+2L2L3C3)$$

$$COS(\theta 2) = \sqrt{1 - SIN^2(\theta 2)}$$

$$COS(\theta 5) = C23.Z_Z+S23(Z_X S1 - Z_Y C1)$$

Le signe est déterminé d'après un critère de continuité

$$COS(\theta 4) = \{(C23(Z_X S1-Z_Y C1)-Z_Z S23) / \sqrt{1-\cos^2(\theta 5)}\} \text{ signe } (\theta 5)$$

$$SIN(\theta 4) = \{(Z_X C1+Z_Y S1) / \sqrt{1 - Cos^2(\theta 5)}\} \text{ signe } (\theta 5)$$

$$SIN(\theta 6) = ((Y_X S1-Y_Y C1)C23 - Y_Z S23)S4 - (Y_X C1+Y_Y S1)C4$$

$$COS(\theta 6).S5 = (Y_X S1-Y_Y C1)S23 + Y_Z C23$$

$$COS(\theta 6).C5 = ((Y_Y C1-Y_X S1)C23 + Y_Z S23)C4 - (Y_X C1+Y_Y S1)S4$$

Lorsque

$S5 = 0$ et $C5 = 1$ nous avons

$$S46 = -Y_X C1 - Y_Y S1$$

$$C46 = (Y_Y C1 - Y_X S1)C23 + Y_Z S23$$

Calcul des angles de représentation de l'effecteur

La matrice de rotation globale $\phi(\Phi_1,\Phi_2,\Phi_3)$ s'écrit :

$$\begin{bmatrix} C1C3+S1S2S3 & -C1S3+S1S2C3 & S1C2 \\ C2S3 & C2C3 & -S2 \\ -S1C3+C1S2S3 & S1S3+C1S2C3 & C1C2 \end{bmatrix}$$

D'où :

$$\hat{S}_1 = S1.\text{signe}\ (C2) = \frac{Z_X}{\sqrt{1 - Z_Y^2}}$$

$$\hat{C}_1 = C1.\text{signe}\ (C2) = \frac{Z_Z}{\sqrt{1 - Z_Y^2}}$$

Deux solutions : $\hat{\Phi}_1(\hat{C}_1,\hat{S}_1)$ et $\hat{\Phi}_1 - \text{signe}\ (\hat{\Phi}_1).\Pi$

Par ailleurs nous avons :

$$\hat{S}_3 = S3.\text{signe}\ (C2) = \frac{X_Y}{\sqrt{1 - Z_Y^2}}$$

$$\hat{C}_3 = C3.\text{signe}\ (C2) = \frac{Y_Y}{\sqrt{1 - Z_Y^2}}$$

De même nous avons deux solutions :

$$\hat{\Phi}_3(\hat{C}_3,\hat{S}_3) \quad \text{et} \quad \hat{\Phi}_3 - \text{signe}\ (\hat{\Phi}_3).\Pi$$

Le signe de C2 est déterminé d'après un critère de continuité, nous avons :

$$\Phi_2(\text{signe}\ (C2) \cdot \sqrt{1 - Z_Y^2}\ ,\ -Z_Y)$$

Lorsque $\Phi_2 = \pm \Pi$ nous avons deux possibilités

$$\{(\pm S1, \pm C1), (S2, \pm C2)\ ;\ (\pm S3, \pm C3)\}$$

On calcule $(\Phi_1 + \Phi_3)$ ou $(\Phi_1 - \Phi_3)$ et on procéde selon I.3.2.

ANNEXE 2

CALCUL DU VECTEUR DE FORCES POUR LE M.A.23

Pour être adapté aux mécanismes des mouvements élémentaires, le vecteur $F/R6 = (f_X, f_Y, f_Z)^t$ est calculé dans le référentiel de l'organe terminal. Le principe de notre calcul consiste à diviser le robot en deux sous-systèmes : S1 (les trois premiers degrés de liberté) et S2 (les trois derniers). Pour obtenir une meilleure sensibilité vis-à-vis de l'effort extérieur nous calculerons les composantes f_X et f_Y d'après S2. Dans le cas où S2 est dans une configuration singulière ces informations sont obtenues d'après S1.

La composante f_Z est calculée d'après S1.

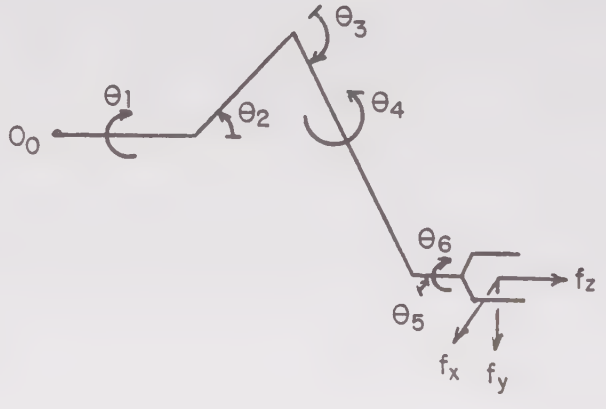
Le vecteur F/R6 s'écrit par rapport à S1 :

$$F/R6 = (M_o^6)^t . J_{O_4}^{-t} . \Gamma_{1,2,3}$$

où $\Gamma_{1,2,3}$ est le vecteur des couples moteurs de S1

J_{O_4} est la matrice jacobienne associée à $O_4 = f(\theta_1, \theta_2, \theta_3)$

M_o^6 est la matrice d'orientation de l'organe terminal.



On retient la composante f_Z d'après ce calcul. D'autre part, dans S2, les incréments Δx et Δy, par rapport à R6, s'écrivent en fonction de $\Delta\theta 4$ et $\Delta\theta 5$:

$$\begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} X^t \frac{\partial Z}{\partial \Theta 4} & X^t \frac{\partial Z}{\partial \Theta 5} \\ Y^t \frac{\partial Z}{\partial \Theta 4} & Y^t \frac{\partial Z}{\partial \Theta 5} \end{bmatrix} \begin{bmatrix} \Delta \Theta 4 \\ \Delta \Theta 5 \end{bmatrix} = A. \Delta \Theta_{4,5}$$

d'où l'expression de f_X et f_Y :

$$\begin{bmatrix} f_X \\ f_Y \end{bmatrix} = A^{-t} \begin{bmatrix} \Gamma_4 \\ \Gamma_5 \end{bmatrix}$$

ANNEXE 3

OBTENTION DE MODELES DE TRANSFERT SIMPLES

L'étude de la chaîne composée d'un moteur électrique, réducteurs à câbles et articulation a conduit (COIFFET, LIEGEOIS, DOMBE et RIVES (42)) à l'établissement du modèle de transmission suivant :

$$\psi \{J_M P^2 + (f_1+f_3)\ p+K\ \} - N(f_3 P + K)\Theta \quad = \quad C$$

$$\Theta \{J_A P^2 + (f_2+N^2 f_3)P + N^2 K\} - N(f_3 P + K)\psi \quad = \gamma$$

où :

ψ : angle moteur ; Θ angle articulation
f_1 : frottement visqueux sur l'arbre du moteur
f_2 : " " " " de l'articulation
f_3 : " " " " dans le réducteur
J_M : inertie sur l'arbre du moteur
J_A : " " " de l'articulation
K : coefficient d'élasticité du câble
N : rapport de réduction
C : couple moteur
γ : couple agissant sur l'articulation

La fonction de transfert liant l'angle moteur ψ aux couples : moteur C et extérieur γ s'écrit :

$$\psi = \frac{1}{H_O} \{H_A C + H_{MA}\gamma\}$$

où $H_A = J_A P^2 + N^2 f_3 P + N^2 K$

$H_{MA} = N(f_3 P + K)$

$$\text{et } H_o = P\{J_M J_A P^3 + (J_A(f_1+f_3)+J_M(f_2+N^2f_3)P^2 + ((J_A+N^2J_M)K$$

$$+f_1f_2 + N^2f_1f_3 + f_2f_3)P + K(f_2+N^2f_1)\}$$

Le polynome $H_o(P)$ admet une décomposition simple lorsque la condition suivante est vérifiée :

$$f_1 << f_3 \quad \text{et} \quad f_2 << N^2 f_3$$

Dans ce cas $H_o(P)$ se décompose en deux facteurs :

$$H_o = \{(J_A+N^2J_M)P + N^2f_1 + f_2\} \cdot \{\frac{J_M J_A}{J_A+N^2J_M} P^2 + f_3 P + K\} \cdot P$$

La fonction de transfert entre l'angle moteur ψ et le couple moteur C est obtenue de la façon suivante (division selon les puissances croissantes à l'ordre deux) :

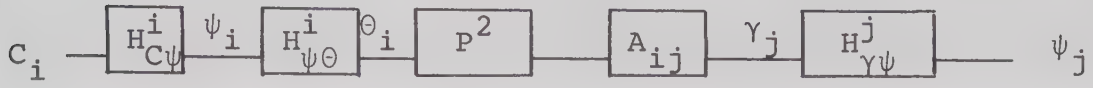
$$\frac{\psi}{C} = H_{C\psi} = \frac{1 + \frac{1}{K}\frac{(J_A/N^2)^2}{J_A/N^2+J_M} P^2}{P\{(\frac{J_A}{N^2} + J_M)P + f_1 + f_2/N^2\}}$$

D'autre part, les fonctions de transfert $H_{\gamma\psi}$ et $H_{\psi\Theta}$ s'écrivent :

$$\frac{\psi}{\gamma} = H_{\gamma\psi} = \frac{1}{P\{(\frac{J_A}{N^2} + J_M)P + f_1 + f_2/N^2\}\{1+\frac{1}{K}\frac{J_M J_A}{J_A+N^2J_M} P^2\}N}$$

$$\frac{\Theta}{\psi} = H_{\psi\Theta} = \frac{1}{N(1 + \frac{1}{K}\frac{J_A}{N^2} P^2)}$$

En présence d'une seule excitation motrice C_i (les autres couples sont constants pour équilibrer la chaîne sur la configuration Θ) le schéma de réaction, sur le jième élément, est le suivant :



En effet, le couple d'excitation sur le jième élément opère au niveau de l'articulation et s'écrit :

$$\gamma_j = A_{ij} \overset{\circ\circ}{\Theta}_i$$

Par conséquent, la fonction de transfert entre C_i et ψ_j est de la forme :

$$\frac{\psi_j}{C_i} = H^{ij}_{C\psi} = \underbrace{H^i_{C\psi} \cdot H^i_{\psi\Theta}}_{\text{fonction de } A_{ii}} \cdot P^2 \cdot A_{ij} \cdot \underbrace{H^j_{\gamma\psi}}_{\text{fonction de } A_{jj}}$$

Pour identifier, il est utile d'engendrer des formes de couples d'excitation C(t) de telle façon que l'on obtienne :

$$\overset{\circ\circ}{C}(t) = 0$$

L'identification est alors effectuée dans le domaine temporel selon un critère quadratique sur les erreurs de mesure. L'estimée A_{ii} s'obtient aisément en fonction des corrélations de grandeurs mesurables :

$$A_{ii}(\Theta) = \{\sum_K \overset{\circ\circ}{\psi} \quad (C - q(\Theta)/N) - (f_1 + f_2/N^2) \sum_K \overset{\circ\circ}{\psi} \overset{\circ}{\psi}\} / \sum_K \overset{\circ\circ}{\psi}^2$$

De la même manière on obtient l'estimée $A_{ij}(\Theta)$:

$$A_{ij}(\Theta) = N \left\{ \frac{f_1+f_2/N^2}{K} \; \frac{J_M J_A}{J_A+N^2 J_M} \sum_1 \overset{\circ}{\Theta} \overset{\circ\circ}{\psi} + (J_M+J_A/N^2) \sum_1 \overset{\circ}{\Theta} \overset{\circ}{\psi} \right.$$

$$\left. + (f_1 + f_2/N^2) \sum_1 \overset{\circ}{\Theta} \psi \right\} / \sum_1 \overset{\circ}{\Theta}^2$$

ANNEXE 4

CALCUL DES ENERGIES CINETIQUE ET POTENTIELLE

Chaque solide S_i de la chaîne articulée est caractérisé par les grandeurs suivantes :

$I_{(3,3)}$: la matrice d'inertie associée au solide S_i
M_i : la masse
D_i : le vecteur du centre de masse par rapport à R_i
L_i : le vecteur de l'extrémité du segment par rapport à R_i

Nous calculerons les vecteurs vitesse de rotation ω_i et de translation V_i et l'énergie cinétique E_i associés au solide S_i. L'énergie totale E du système s'écrit :

$$E = \sum_{i=1}^{n} E_i$$

Elle peut s'exprimer sous la forme suivante :

$$E = \sum_{i=1}^{n} \sum_{j=i}^{n} a_{ij} . \dot{\theta}_i \, \dot{\theta}_j$$

Les coefficients d'inertie se déduisent de E :

$$A_{ij} = \begin{cases} a_{ii} \text{ si } j = i \\ \frac{1}{2} a_{ij} \text{ sinon} \end{cases}$$

Nous développerons les expressions de ω_i, V_i et E_i :

Calcul de ω_1, V_1 et E_1

$$\omega_1 = \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_1 \end{bmatrix} \quad ; \quad V_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad ; \quad 2E_1 = IZ1\ \dot{\theta}_1^2$$

Calcul de ω_2, V_2 et E_2

$$\omega_2 = \begin{bmatrix} \dot{\theta}_2 \\ S2\dot{\theta}_1 \\ C2\dot{\theta}_1 \end{bmatrix} \quad ; \quad V_2 = \begin{bmatrix} S2d2\dot{\theta}_1 \\ -d2\dot{\theta}_2 \\ 0 \end{bmatrix} \quad ; \quad 2E_2 = \dot{\theta}_1^2 \{M2d_2^2S^2 2 + IY2S^2 2 + IZ2C^2 2\} + \dot{\theta}_2^2 \{M2d_2^2 + IX2\}$$

Calcul de ω_3, V_3 et E_3

$$\omega_3 = \begin{bmatrix} \dot{\theta}_2 + \dot{\theta}_3 \\ S23\dot{\theta}_1 \\ C23\dot{\theta}_1 \end{bmatrix} \quad ; \quad V_3 = \begin{bmatrix} S21_2\dot{\theta}_1 + S23\ d_3\dot{\theta}_1 \\ -C31_2\dot{\theta}_2 - d_3(\dot{\theta}_3 + \dot{\theta}_2) \\ S31_2\dot{\theta}_2 \end{bmatrix}$$

$$2E3 = \dot{\theta}_1^2 \{M3\ (S^2 21_2 + S^2 23d_3^2 + 2S2S231_2d_3) + IY3S^2 23 + IZ3C^2 23\}$$

$$+ \dot{\theta}_2^2 \{M3\ (C^2 31^2_2 + d_3^2 + 2C31_2d_3 + S^2 31^2 2) + IX3\}$$

$$+ \dot{\theta}_3^2 \{M3d_3^2 + IX3\} + \dot{\theta}_2\dot{\theta}_3 \{2M3d_3^2 + 2C31_2d_3 + 2IX3\}$$

Calcul de ω_4, V_4 et E_4

$$\omega_4 = \begin{bmatrix} C4(\dot{\theta}_2 + \dot{\theta}_3) + S23S4\dot{\theta}_1 \\ -S4(\dot{\theta}_2 + \dot{\theta}3) + S23C4\dot{\theta}_1 \\ C23\ \dot{\theta}_1 + \dot{\theta}_4 \end{bmatrix}$$

$$V_4 = \begin{bmatrix} C4(S2l_2\dot{\theta}_1+S23l_3\dot{\theta}_1)-S4(C3l_2\dot{\theta}_2+l_3(\dot{\theta}_2+\dot{\theta}_3))+d_4(-S4(\dot{\theta}_2+\dot{\theta}_3)+S23C4\dot{\theta}_1) \\ -S4(S2l_2\dot{\theta}_1+S23l_3\dot{\theta}_1)-C_4(C3l_2\dot{\theta}_2+l_3(\dot{\theta}_2+\dot{\theta}_3))-d_4(C4(\dot{\theta}_2+\dot{\theta}_3)+S23S4\dot{\theta}_1) \\ S3l_2\dot{\theta}_2 \end{bmatrix}$$

$$2E4 = \dot{\theta}_1^2 \{M4(S^2 21^2 + S^2 231^2 + 2S2S231_2 l_{34}) + IX4S^2 23S^2 4 + IY4S^2 23C^2 4 + IZ4C^2 23\}$$

$$+ \dot{\theta}_2^2 \{M4(C^2 31^2_2 + l^2_2 + 2C3l_2 l_{34}) + IX4C^2 4 + IY4S^2 4\}$$

$$+ \dot{\theta}_3^2 \{M4l^2_{34} + IX4C^2 4 + IY4S^2 4\} + \dot{\theta}_4^2 \{IZ4\}$$

$$+ \dot{\theta}_1\dot{\theta}_2 \{2IX4S23C4S4 - 2S23C4S4IY4\}$$

$$+ \dot{\theta}_1\dot{\theta}_3 \{2IX4S23C4S4 - 2S23C4S4IY4\}$$

$$+ \dot{\theta}_2\dot{\theta}_3 \{2M4l^2_{34} + 2M4C3l_2 l_{23} + 2IX4C^2 4 + 2IY4S^2 4\}$$

$$+ \dot{\theta}_1\dot{\theta}_4 \{2IZ4C23\}$$

Calcul de ω_5, V_5 et E_5

$$\omega_5 = \begin{bmatrix} C4(\dot{\theta}_2+\dot{\theta}_3)+S23S4\dot{\theta}_1+\dot{\theta}_5 \\ C5(-S4(\dot{\theta}_2+\dot{\theta}_3)+S23C4\dot{\theta}_1)+S5(C23\dot{\theta}_1+\dot{\theta}_4) \\ -S5(-S4(\dot{\theta}_2+\dot{\theta}_3)+S23C4\dot{\theta}_1)+C5(C23\dot{\theta}_1+\dot{\theta}_4)+\dot{\theta}_6 \end{bmatrix}$$

$$V_5 = \begin{bmatrix} C4(S2l_2\dot{\theta}_1+S23l_{34}\dot{\theta}1)-S4(C3l_2\dot{\theta}2+l_{34}(\dot{\theta}_2+\dot{\theta}_3))+d_5(C5(-S4(\dot{\theta}_2+\dot{\theta}_3)+S23C4\dot{\theta}_1)+S5(C23\dot{\theta}_1+\dot{\theta}_4)) \\ C5(-S4(S2l_2\dot{\theta}_1+S23l_{34}\dot{\theta}_1)-C4(C3l_2\dot{\theta}_2+l_{34}(\dot{\theta}_2+\dot{\theta}_3)))+S5S3l_2\dot{\theta}_2-d_5(C4(\dot{\theta}_2+\dot{\theta}_3)+S23S4\dot{\theta}_1+\dot{\theta}_5) \\ -S5(-S4(S2l_2\dot{\theta}_1+S23l_{34}\dot{\theta}1)-C4(C3l_2\dot{\theta}2+l_{34}(\dot{\theta}_2+\dot{\theta}_3)))+C5S3l_2\dot{\theta}_2 \end{bmatrix}$$

$$2E_5 = \dot{\theta}_1^2 \{S^2 23S^2 4IX5 + S^2 23C^2 4C^2 5IY5 + 2C23S23C4C5S5IY5 + S^2 23C^2 4S^2 5IZ5 + C^2 23C^2 5IZ5 + C^2 23S^2 5IY5 - 2S23C23C5S5C4IZ5 + M5(S^2 21^2_2$$

$$+ S^2_{23}1^2_{34}+2S2S23l_2l_{34}+S^2_{23}C^2_4C^2_5d^2_5+C^2_{23}S^2_5d^2_5+S^2_{23}S^2_4d^2_5$$
$$+2S2S23C5l_2d_5+2C23S23C4C5S5d^2_5+2S2C4C23S5d_5l_2+2C23S23C4S5$$
$$d_5l_{34})\}$$

$$+\overset{\circ}{\theta}{}^2_2 \{C^2_4IX5+S^2_4C^2_5IY5+M5(l^2_2+l^2_{34}+2C3l_2l_{23}+S^2_4C^2_5d^2_5$$
$$+C^2_4d^2_5+2C3C5l_2d_5+2C5l_{34}d_5-2S3C4S5l_2d5)\}$$

$$+\overset{\circ}{\theta}{}^2_3 \{C^2_4IX5+C^2_5S^2_4IY5+S^2_5S^2_4IZ5+M5(l^2_{34}+S^2_4C^2_5d^2_5+C^2_4d^2_5$$
$$+2C5d_5l_{34})\}$$

$$+\overset{\circ}{\theta}{}^2_4 \{S^2_5IY5+C^2_5IZ5+S^2_5M5d^2_5\}$$

$$+\overset{\circ}{\theta}{}^2_5 \{IX5+M5d^2_5\} + \overset{\circ}{\theta}{}^2_6 \{IZ5\}$$

$$+2\overset{\circ}{\theta}_1\overset{\circ}{\theta}_2 \{S23C4S4IX5+C23S4C5S5IY5-S23C4S4S^2_5IZ5$$
$$-C23S4C5S5IZ5-S23C4S4C^2_5IY5-M5(S23C4S4C^2_5d^2_5$$
$$+C23S4C5S5d^2_5-S23C4S4d^2_5+C23S4S5d_5l34 +$$
$$S3S23S4S5l_2d_5+C2S4S5l_2d_5)\}$$

$$2\overset{\circ}{\theta}_1\overset{\circ}{\theta}_3 \{S23C4S5IX5-C23S4C5S5IY5-S23C4S4S^2_5IZ5+C23S4C5S5IZ5$$
$$-S23C4S4C^2_5IY5-M5(C^2_5S4C4S23d^2_5+C23S4C5S5d^2_5-S23C4S4d^2_5$$
$$+C23S4S5d_5l_{34})\}$$

$$2\overset{\circ}{\theta}_1\overset{\circ}{\theta}_4 \{C23S^2_5(IY5+IZ5)+S23C4C5S5(IY5-IZ5)+M5(S^2_5C23d^2_5)$$
$$+C5S5S23C4d^2_5+S2C4S5l_2d_5+S23C4S5d_5l_{34})\}$$

$$2\overset{\circ}{\theta}_1\overset{\circ}{\theta}_5 \{S23_54IX5+M5(S23S4d^2_5+S2S4C5l_2d_5+S23S4C5d_5l_{34})\}$$

$$2\overset{\circ}{\theta}_2\overset{\circ}{\theta}_3 \{C^2_4IX5+S^2_4C^2_5IY5+S^2_4S^2_5IZ5+M5(l^2_{34}+C3l_2l_{34}+C^2_4d^2_5-$$
$$S^2_4C^2_5d^2_5+C3C5d_5l_2+2C5d_5l_{34}-S3C4S5l_2d_5)\}$$

$$2\overset{\circ}{\theta}_2\overset{\circ}{\theta}_4 \{S4C5S5(IZ5-IY5)-M5(S4C5S5d^2_5+C3S4S5l_2d_5+S4S5d_5l_{34})\}$$

$$2\overset{\circ}{\theta}_3\overset{\circ}{\theta}_4 \{S4C5S5(IZ5-IY5)-M5(C5S5S4d^2_5)\}$$

$$2\overset{\circ}{\theta}_3\overset{\circ}{\theta}_5 \{C4IX5+M5(C4d^2_5+C4C5d_5l_{34})\}$$

$$2\overset{\circ}{\theta}_2\overset{\circ}{\theta}_5 \{C4IX5+M5(C4d^2_5+C3C4C5l_2d_5+C4C5l_{34}d_5-S3S5l_2d_5)\}$$

Calcul de E_6

$$2E_6 = \dot{\theta}_1^2 \{S^2_{23}IY6+C^2_{23}IZ6+M6(S^2_{21_6}+2S2S23l_6d_6+S^2_{23}d^2_6)\}$$

$$\dot{\theta}_2^2 \{IX6+M6d^2_6+l^2_6+2C3l_6d_6\}$$

$$\dot{\theta}_3^2 \{IX6+M6d^2_6\}$$

$$2\dot{\theta}_2\dot{\theta}_3 \{IX6+M6(C3l_6d_6+d^2_6)\}$$

La ième composante de la pesanteur Q_i s'écrit en fonction des dérivées partielles des composantes verticales des centres de masse :

$$Q_i = \sum_{j=1}^{n} gM_j \frac{\partial Y_j}{\partial \theta_i}$$

Pour le M.A.23 les masses M_1, M_2, M_3 et M_4 sont compensées par des contrepoids, la composante Y_5 est la seule à prendre en compte :

$$Y_5 = -C1S2l_2-C1S23(l_3+l_4)+(S1S4S5-C1C23C4S5-C1S23C5)D5$$

Nous déduisons les composantes Q_i :

$$Q_1 = gM5 \{S1S2l_2+S1S23(l_3+l_4)+D5(C1S4S5+S1C23C4S5+S1S23C5)\}$$

$$Q_3 = gM5 \{-C1C23(l_3+l_4)+(C1S23C4S5-C1C23C5)D5\}$$

$$Q_2 = gM5 \{-C1C2l_2\} + Q_3$$

$$Q_4 = gM5D5 \{S1C4S5+C1C23S4S5\}$$

$$Q_5 = gM5D5 \{S1S4C5-C1C23C4C5+C1S23S5\}$$

```
!PL/M-86 COMPILER     NTR23                                                    20/12/81

ISIS-II PL/M-86 V2.1 COMPILATION OF MODULE NTR23
OBJECT MODULE PLACED IN :F1:NTR23.OBJ
COMPILER INVOKED BY:  PLM86 :F1:NTR23.SRC DEBUG LARGE OPTIMIZE(3) DATE(20/12/81)


   1          NTR23:DO;
              $NOLIST INCLUDE(:F1:PLMLIB.SRC)

                              /*****************************************************************/
                              /*                                                               */
                              /*          SYSTEME  DE  COORDINATION  DES  MOUVEMENTS           */
                              /*                                                               */
                              /*                    MANIPULATEUR M.A. 23                       */
                              /*                                                               */
                              /*****************************************************************/
```

```
21   1          DCL CONTROL$87 LIT'033EH';
22   1          DCL EPA LIT'2094H',EPB LIT'2095H',EPC LIT'2096H',EPCD LIT'2097H';
23   1          DCL SPA LIT'2090H',SPB LIT'2091H',SPC LIT'2092H',SPCD LIT'2093H';
24   1          DCL EMA LIT'20D0H',EMB LIT'20D1H',EMC LIT'20D2H',EMCD LIT'20D3H';/*MITRA VERS MICRO*/
25   1          DCL SMA LIT'20D4H',SMB LIT'20D5H',SMC LIT'20D6H',SMCD LIT'20D7H';
26   1          DCL MASK$LEC LIT'08H',MASK$CONV LIT'10H',MASK$DECLT LIT'80H',MASK9 LIT'0FH';
27   1          DCL ARM1 LIT'0DH',ARM2 LIT'09H';                    /*ARMEMENT DES IT */
28   1          DCL MASK$ECRI LIT'08H',MASKC LIT'8000H',MASK$FON LIT'07FF0H';
29   1          DCL MASK$NBA LIT'0FH',READY LIT'80H';
30   1          DCL TABARG(16) WORD;
31   1          DCL (FON,EN) WORD;
32   1          DCL (J,ADR,M,N,K,I)     BYTE;
33   1          DCL INI1 LIT'0A0H',INI3 LIT'0BBH',INI5 LIT'99H',INI7 LIT'80H';
34   1          DCL (L,ER) BYTE;
35   1              DCL INDEX WORD;
36   1          DCL IN WORD;
37   1          DCL (X,Y,A) REAL;
38   1          DCL W(51) REAL;
39   1                  DCL (C1,S1,C2,S2,C3,S3,C23,S23,C4,S4,C5,S5,C6,S6) REAL;
40   1              DCL RR(14) REAL DATA(3141.59,6.17,6.07,6.,3.08,3.135,3.158,2047.,140.,1.57079632
                                        1000.,1.,1570.796);
41   1          DCL TABCOS (160) REAL DATA (1.,
                0.999950001,0.999800007,0.999550034,0.999200107,0.99875026,
                0.99820054,0.997551,0.996801706,0.995952733,0.995004165,etc
42   1          DCL TABATAN(200) REAL DATA (0.,
                0.012499349E+03,0.024994794E+03,0.037482437E+03,0.049958396E+03,0.06241881E+03,
                0.074859848E+03,0.08727713E+03,0.0996685653E+03,0.112028962E+03,0.124354995E+03, etc
43   1          DCL TABARCS(85) REAL DATA (0.,
                12.500325,25.00260483,37.508795,50.020857,62.540762, etc
44   1          DCL TABRACA (93) REAL DATA (1.,
                0.99995,0.9998,0.9995499,0.9991997,0.9987492,0.9981984,0.997547,0.9967949,0.9959418,0.9etc
45   1          DCL TABRACB(68) REAL DATA (0.4358899,
                0.4326489,0.4293777,0.4260757,0.4227422,0.4193763,0.4159773,0.4125444,0.4090757,0.4055?etc
46   1                  DCL PL1(3) REAL;
47   1                  DCL PL2(3) REAL;
48   1                  DCL PL3(3) REAL;
49   1                  DCL TR(4) REAL;
50   1                  DCL ROT(10) REAL;
```

```
            /*****************************************************************************/
            /*                                                                           */
            /*                          SOUS PROGRAMME DE CALCUL DE RAC(1-X**2)          */
            /*                                                                           */
            /*****************************************************************************/

51   1          RACS:   PROC(X) REAL;
52   2              DCL I WORD;
53   2              DCL (X,Y,VAL) REAL;
L/M-86 COMPILER    NTR23                                                    20/12/81


54   2              IF X=0. THEN RETURN 1.;
56   2              VAL=ABS(X);
57   2              IF VAL>=0.9 THEN DO;
59   3                  VAL=(VAL-0.9)*640.;
60   3                  I=LOW(UNSIGN(FIX(VAL-0.5)));
61   3                  Y=TABRACB(I)+(VAL-FLOAT(SIGNED(I)))*(TABRACB(I+1)-TABRACB(I));
62   3                  END;
63   2              ELSE DO;
64   3                  VAL=VAL*100.;
65   3                  I=LOW(UNSIGN(FIX(VAL-0.5)));
66   3                  Y=TABRACA(I)+(VAL-FLOAT(SIGNED(I)))*(TABRACA(I+1)-TABRACA(I));
67   3                  END;
68   2              RETURN Y;
69   2              END RACS;
            /*****************************************************************************/
            /*                                                                           */
            /*                          SOUS      PROGRAMME FONCTION ARC SINUS           */
            /*                                                                           */
            /*****************************************************************************/

70   1          ARCS:   PROC(X) REAL;
71   2              DCL I WORD;
72   2              DCL (X,Y,VAL) REAL;
73   2              IF X=0. THEN RETURN 0.;
75   2              VAL=ABS(X*80.);
76   2              I=LOW(UNSIGN(FIX(VAL-0.5)));
77   2              IF X>=0. THEN Y=TABARCS(I)+(VAL-FLOAT(SIGNED(I)))*(TABARCS(I+1)-TABARCS(I));
79   2                  ELSE Y=-TABARCS(I)-(VAL-FLOAT(SIGNED(I)))*(TABARCS(I+1)-TABARCS(I));
80   2              RETURN Y;
81   2              END ARCS;
```

```
               /*****************************************************************************/
               /*                                SOUS PROGRAMME FONCTION COS                 */
               /*****************************************************************************/

  82   1           COS:    PROC (X)REAL;
  83   2               DCL I WORD;
  84   2               DCL (X,Y,VAL) REAL;
  85   2               IF X=0. THEN RETURN 1.;
  87   2               VAL=ABS(X*100.);
  88   2               I=UNSIGN(FIX(VAL-0.5));
  89   2                Y=TABCOS(I)+(VAL-FLOAT(SIGNED(I)))*(TABCOS(I+1)-TABCOS(I));
  90   2               RETURN Y;
  91   2               END COS;
               /*******************************************************************************/
               /*                                   SOUS PROGRAMME FONCTION ARC TANGENTE        */
               /*******************************************************************************/

  92   1           ATAN:   PROC(X)REAL;
  93   2               DCL I WORD;
  94   2               DCL (X,Y,VAL) REAL;
  95   2               IF X=0. THEN RETURN 0.;
  97   2               VAL=ABS(X*80.);
  98   2               I=UNSIGN(FIX(VAL-0.5));
  99   2               IF X>=0. THEN Y=TABATAN(I)+(VAL-FLOAT(SIGNED(I)))*(TABATAN(I+1)-TABATAN(I));
 101   2                   ELSE Y=-TABATAN(I)-(VAL-FLOAT(SIGNED(I)))*(TABATAN(I+1)-TABATAN(I));
 102   2               RETURN Y;
 103   2               END ATAN;
PL/M-86 COMPILER    NTR23                                                                        20/12/81


               /*******************************************************************/
               /*                           LECPOTARS                             */
               /*******************************************************************/
 104   1                        LEC$POTARS: PROC(N,ADR);
 105   2                             DCL (N,O,ADR,P,I) BYTE;
 106   2                             DO O=1 TO N;
 107   3                             OUTPUT(EPB)=ADR;
                               /*    ATTENTE D'ADRESSAGE    */
 108   3                             P=0;P=0;P=0;P=0;
                               /* DECLENCHEMENT DE LA CONVERSIN  */
 112   3                             OUTPUT(EPB)=ADR OR MASK$DECLT;
```

```
113   3                              OUTPUT(EPB)=ADR;
114   3                              ADR=ADR+1;
115   3                              IF P=1 THEN;
117   3                              DO WHILE (INPUT(SMC) AND MASK$LEC)=0;
118   4                              END;
119   3                              P=1;
                                  /* TEST DE FIN DE CONVERSION      */
120   3                              DO WHILE (INPUT(EPA) AND MASK$CONV)<>0;
121   4                              END;
                                  /*   SORTIE DES DONNEES    */
122   3                              OUTPUT(SMB)=INPUT(EPC);
123   3                              OUTPUT(SMA)=INPUT(EPA) AND MASK9;
124   3                              END;
                                  /* TEST LEC                       */
125   2                             DO WHILE (INPUT(SMC) AND MASK$LEC)=0;
126   3                              END;
127   2                               END LEC$POTARS;
            /*****************************************************************/
            /*                    EPX EPY EPZ EPXYZ ROTI                     */
            /*****************************************************************/
128   1                              EPX: PROC(ARG);
129   2                              DCL ARG REAL;
130   2                              W(41)=W(1)+ARG*(W(8)*W(12)-W(11)*W(9));
131   2                              W(42)=W(2)+ARG*(W(10)*W(9)-W(7)*W(12));
132   2                              W(43)=W(3)+ARG*(W(7)*W(11)-W(10)*W(8));
133   2                              END EPX;
134   1                             EPY: PROC(ARG);
135   2                              DCL ARG REAL;
136   2                              DCL I BYTE;
137   2                              DO I=1 TO 3;
138   3                              W(I+40)=W(I)+ARG*W(I+6);
139   3                              END;
140   2                              END EPY;
141   1                             EPZ: PROC(ARG);
142   2                              DCL ARG REAL;
143   2                              DCL I BYTE;
144   2                              DO I=1 TO 3;
145   3                              W(I+40)=W(I)+ARG*W(I+9);
146   3                              END;
147   2                              END EPZ;
```

```
148   1                    EPXYZ :PROC;
149   2                       DCL I BYTE;
150   2                       DO I=1 TO 3;
151   3                       W(I)=W(I+40);
152   3                        END;
PL/M-86 COMPILER    NTR23                                                  20/12/81

153   2                       END EPXYZ;
154   1                    ROTI: PROC;
155   2                        DCL I BYTE;
156   2                        DO I=4 TO 6;
157   3                        W(I+40)=W(I);
158   3                        END;
159   2                        END ROTI;
          /************************************************************************/
          /*                                   TRANSFORMATEUR                      */
          /************************************************************************/
184   1               TRANSF:PROC(AIN,ADW);
185   2                    DCL (AIN,ADW) POINTER;
186   2                    DCL W BASED ADW(1) REAL;
187   2                    DCL IN BASED AIN WORD;
188   2                    DCL I BYTE;
189   2                    DCL (X5,Y5,Z5,A,B,D) REAL;
190   2                    DCL (T1,T2,T3,T4,T5,T6) REAL;
191   2                    X5=W(1)-RR(8)*W(10);
192   2                    Y5=W(2)-RR(8)*W(11);
193   2                    Z5=W(3)-RR(8)*W(12);
194   2                    IF ABS(Y5)<+0.01 THEN DO;IN=6;RETURN;END;
199   2                    B=X5*X5+Y5*Y5+Z5*Z5;
200   2                    IF B>1.E+06 THEN DO;IN=6;RETURN;END;
205   2                    A=-X5/Y5;
206   2                    IF ABS(A)>+2.1446 THEN DO;IN=6;RETURN;END;;
212   2                    T1=ATAN(A);
213   2                    C1=COS(T1/RR(11));
214   2                    S1=C1*A;
215   2                    C3=(B-586708.)/524208.;
216   2                    T3=ARCS(C3)-RR(13);
217   2                    IF T3<-2356. THEN DO;T3=-2356.;C3=-0.707;END;
222   2                    ELSE DO;IF T3>-523. THEN DO;T3=-523.;C3=0.866;END;END;
```

```
229   2                          S3=-RACS(C3);
230   2                          S2=-(Y5*(402.+652.*C3)/C1+Z5*S3*652.)/
                                     (586708.+524208.*C3);
231   2                          T2=ARCS(S2);
232   2                    IF T2<-1221. THEN DO;T2=-1221.;S2=-0.94;END;
```



```
237   2                    ELSE DO;IF T2>994. THEN DO;T2=994.;S2=0.838;END;END;
244   2                          C2=RACS(S2);
245   2                          S23=C2*S3+S2*C3;
246   2                          C23=C2*C3-S2*S3;
247   2                          A=S1*W(10)-C1*W(11);
248   2                          C5=C23*W(12)+S23*A;
249   2                    IF C5>0.997 THEN DO;CALL SING(@T4,@T5,@T6);
252   3                                      GOTO ETQ;END;
254   2                    T5=RR(13)-ARCS(C5);
255   2                    IF T5>2180. THEN DO;T5=2180.;C5=-0.572;END;
260   2                    ELSE DO;IF T5<-600. THEN DO;T5=-600.;C5=0.82;END;END;
267   2                          S5=RACS(C5);
268   2                          C4=(C23*A-S23*W(12))/S5;
269   2                          S4=(C1*W(10)+S1*W(11))/S5;
270   2                    IF C4>=0. THEN T4=ARCS(S4);
272   2                    ELSE DO;T4=RR(13)-ARCS(C4);
274   3                            IF S4<0. THEN T4=-T4;
276   3                            END;
277   2                    IF C5<0.82 THEN GOTO ET;
279   2                    IF T4>=0. THEN DO;T6=T4-RR(0);END;
283   2                    ELSE DO;T6=T4+RR(0);END;
286   2                    IF ABS(T4-W(16))>ABS(T6-W(16)) THEN DO;
288   3                    T5=-T5;T4=T6;S4=-S4;C4=-C4;S5=-S5;END;
294   2                    ET: A=W(7)*S1-W(8)*C1;
295   2                        B=W(7)*C1+W(8)*S1;
296   2                        D=A*C23-W(9)*S23;
297   2                        S6=D*S4-B*C4;
298   2                    IF ABS(C5)>ABS(S5) THEN C6=-(D*C4+B*S4)/C5;
300   2                    ELSE C6=(A*S23+W(9)*C23)/S5;
301   2                    IF C6>=0. THEN T6=ARCS(S6);
303   2                    ELSE DO;T6=RR(13)-ARCS(C6);
305   3                            IF S6<0. THEN T6=-T6;
307   3                            END;
```

```
308   2                          IF T6>3140. THEN DO; T6=3140.;END;
312   2                          ELSE DO;IF T6<-3140. THEN DO;T6=-3140.;END;END;
318   2                          ETQ:CALL DEC(@T4,@T5,@T6);
319   2                          W(13)=T1;W(14)=T2;W(15)=T3;
322   2                          W(16)=T4;W(17)=T5;W(18)=T6;
325   2                          END TRANSF;
      /*****************************************************************/
      /*                          ENVSIXCONS                           */
      /*****************************************************************/
326   1                    ENV$SIX$CONS: PROC;
327   2                          DCL (I,ADR) BYTE;
328   2                          ADR=1;
329   2                          DO I=1 TO 6;
330   3                          OUTPUT(SPA)=NOT HIGH (TABARG(I));
331   3                          OUTPUT(SPB)=NOT LOW (TABARG(I));
332   3                          OUTPUT(SPC)=ADR;
333   3                          ADR=ROL(ADR,1);
334   3                          END;
335   2                          OUTPUT(SPC)=0;
336   2                          END ENV$SIX$CONS;
337   1                    ENV$CONS: PROC(ADR);
338   2                          DCL ADR BYTE;
339   2                          OUTPUT(SPA)=NOT HIGH(TABARG(1));
340   2                          OUTPUT(SPB)=NOT LOW(TABARG(1));
PL/M-86 COMPILER    NTR23                                          20/12/81

341   2                          OUTPUT(SPC)=ADR;
342   2                          OUTPUT(SPC)=0;
343   2                          END ENV$CONS;
      /*****************************************************************/
      /*                            COUPD                              */
      /*****************************************************************/
344   1                      COUPD: PROC;
345   2                        DCL NN WORD;
346   2                        NN=2047;
347   2                        TABARG(1)=NN-UNSIGN(FIX(W(13)));
348   2                        TABARG(2)=NN+UNSIGN(FIX(W(14)));
349   2                        TABARG(3)=4000+UNSIGN(FIX(W(15)));
350   2                        TABARG(4)=NN+UNSIGN(FIX(W(16)))/2;
351   2                        TABARG(5)=1800+UNSIGN(FIX(W(17)));
352   2                        TABARG(6)=NN-UNSIGN(FIX(W(18)))/2;
353   2                        CALL ENVSIXCONS;
354   2                        END COUPD;
```

```
              /*********************************************************************/
              /*                        LECTURES                                   */
              /*********************************************************************/
355   1                         LECTURES: PROC(N,ADR);
356   2                            DCL (N,ADR) BYTE;
357   2                            DCL KO WORD;
358   2                            DCL P BYTE;
359   2                            KO=256;
360   2                             DO I=1 TO N;
361   3                            OUTPUT(EPB)=ADR;
362   3                            P=0;P=0;P=0;P=0;
366   3                            OUTPUT(EPB)=ADR OR MASK$DECLT;
367   3                            OUTPUT(EPB)=ADR;
368   3                            ADR=ADR+1;
369   3                            DO WHILE(INPUT(EPA) AND MASK$CONV) <>0;
370   4                            END;
371   3                            W(I+40)=FLOAT(SIGNED(KO*(INPUT(EPA) AND MASK9)+INPUT(EPC)));
372   3                     END;
373   2                            END LECTURES;
              /*************************************************************************/
              /*                          DELAI ET GAIN                                */
              /*************************************************************************/
374   1                           DELAI: PROC(K);
375   2                              DCL (K,I) BYTE;
376   2                              DCL ARG REAL;
377   2                              DO I=1 TO K;
378   3                              ARG=RR(7)*RR(8);
379   3                              END;
380   2                              END DELAI;
381   1                           GAIN:PROC(ADR);
382   2                              DCL (ADR,ADRS) WORD;
383   2                              DCL I BYTE;
384   2                              ADRS=ADR+20A0H;
385   2                              DO I=1 TO 6;
386   3                              OUTPUT(ADRS)=LOW(TABARG(I));
387   3                              ADRS=ADRS+0003H;
388   3                              END ;
389   2                              END GAIN;
```

```
            /*************************************************************************/
            /*                        TETAP  DEPART                                  */
PL/M-86 COMPILER    NTR23                                                    20/12/81


            /*************************************************************************/
390   1                            TETAP: PROC;
391   2                                DCL I BYTE;
392   2                                CALL LECTURES(6,0H);
393   2                                DO I=1 TO 6;
394   3                                W(I+40)=(W(I+40)-RR(7))/(204.7*RR(I));
395   3                                END;
396   2                                W(31)=3.62*W(41);
397   2                                W(32)=6.*W(42)-0.773;
398   2                                W(33)=5.56*W(43)-9.125*W(42)+3.56;
399   2                                W(34)=3.55*W(44)+1.42*W(43)-0.324;
400   2                                W(35)=3.78*W(45)-1.2*W(43)+4.*W(46)+3.46;
401   2                                W(36)=2.41*(W(45)-W(46))+0.65;
402   2                                DO I=31 TO 36;
403   3                                W(I)=RR(7)+W(I)*204.7;
404   3                                END;
405   2                                W(41)=RR(7)-W(31);
406   2                                W(42)=W(32)-RR(7);
407   2                                W(43)=W(33)-4000.;
408   2                                W(44)=(W(34)-RR(7))*2.;
409   2                                W(45)=W(35)-1800.;
410   2                                W(46)=(RR(7)-W(36))*2.;
411   2                                END TETAP;
412   1                            DEPART: PROC;
413   2                              DCL (J,K) BYTE;
414   2                              DCL I WORD;
415   2                              K=50;
416   2                              CALL TETAP;
417   2                              DO I=1 TO 20;
418   3                              W(I)=0.;
419   3                              END;
420   2                              W(2)=652.;
421   2                              W(3)=542.;
422   2                              W(8)=RR(12);
423   2                              W(12)=W(8);
424   2                              W(15)=-RR(13);
425   2                              W(17)=RR(13);
```

```
426   2                              DO I=1 TO 6;
427   3                              W(I+30)=(W(I+12)-W(I+40))/250.;
428   3                              W(I+12)=W(I+40);
429   3                              END;
430   2                              DO I=1 TO 250;
431   3                              DO J=1 TO 6;
432   4                              W(J+12)=W(J+12)+W(J+30);
433   4                              END;
434   3                              CALL DELAI(K);
435   3                              CALL COUPD;
436   3                              END;
437   2                              W(19)=2431.;TABARG(1)=UNSIGN(FIX(W(19)));CALL ENVCONS(40H);
440   2                              DO I=0 TO 2;TR(I+1)=0.;PL1(I)=W(16);PL2(I)=W(17);PL3(I)=W(18);END;
446   2                              DO I=1 TO 9;ROT(I)=0.;END;
449   2                              END DEPART;
        /*************************************************************/
        /*                          INTER                            */
        /*************************************************************/
450   1                           COSR: PROC(X) REAL;
451   2                                DCL (X,Y) REAL;
PL/M-86 COMPILER    NTR23                                                    20/12/81

452   2                                Y=ABS(X);
453   2                                IF Y<=RR(9) THEN Y=COS(X);
455   2                                ELSE Y=-COS(RR(10)-Y);
456   2                                RETURN Y;
457   2                                END COSR;
458   1                            SINR :PROC(X) REAL;
459   2                                 DCL (X,Y) REAL;
460   2                                 Y=ABS(X);
461   2                                 IF Y<=RR(9) THEN Y=COS(RR(9)-Y);
463   2                                 ELSE Y=COS(Y-RR(9));
464   2                                 IF X<0. THEN RETURN -Y;
466   2                                 RETURN Y;
467   2                                 END SINR;
468   1                            OPERAT: PROC;
469   2                                 DCL I BYTE;
470   2                                 DCL (C1,S1,C2,S2,C3,S3,A) REAL;
471   2                                 W(5)=-ARCS(W(11));A=RACS(W(11));
473   2                                 IF ABS(A)<0.01 THEN RETURN;
475   2                                 S1=W(10)/A;C1=W(12)/A;C3=W(8)/A;
```

```
478   2                                   S3=S1*W(9)-C1*W(7);
479   2                                   IF C1>=0. THEN W(4)=ARCS(S1);
481   2                                   ELSE DO; W(4)=RR(13)-ARCS(C1);
483   3                                        IF S1<0. THEN W(4)=-W(4);END;
486   2                                   IF C3>=0. THEN W(6)=ARCS(S3);
488   2                                   ELSE DO;W(6)=RR(13)-ARCS(C3);
490   3                                       IF S3<0. THEN W(6)=-W(6);END;
493   2                                   DO I=4 TO 6;W(I)=W(I)/RR(11);END;
496   2                                   END OPERAT;
497   1                                INTER : PROC;
498   2                                  DCL (C1,C2,C3,S1,S2,S3,S2C3) REAL;
499   2                                  C1=COSR(W(4)); S1=SINR(W(4));
501   2                                  C2=COSR(W(5)); S2=SINR(W(5));
503   2                                  C3=COSR(W(6)); S3=SINR(W(6));
505   2                                  S2C3=S2*C3;
506   2                                  W(7)=S1*S2C3-C1*S3;
507   2                                  W(8)=C2*C3;
508   2                                  W(9)=S1*S3+C1*S2C3;
509   2                                  W(10)=C2*S1;
510   2                                  W(11)=-S2;
511   2                                  W(12)=C1*C2;
512   2                                  END INTER;
            /*****************************************************************/
            /*                              FILTR                            */
            /*****************************************************************/
513   1                              FILTR: PROC(NTIM);
514   2                                  DCL (NTIM,N1,N2,N3,N4,N5,N6,NN,I) WORD;
515   2                                  DCL J BYTE;
516   2                                  DCL DEL(7) REAL;
517   2                                  DCL TI(7) REAL;
518   2                                  DCL (ALPHA,POS,VIT,ACC) REAL;
519   2                                  CALL OPERAT;
520   2                                  DO I=1 TO 6;
521   3                                  TI(I)=W(I);
522   3                                  DEL(I)=W(I+40)-W(I);
523   3                                  END;
524   2                                  N1=(NTIM/12)+1;
525   2                                  NN=N1*12;
PL/M-86 COMPILER    NTR23                                          20/12/81
```

```
526   2                                 N3=N1*3;
527   2                                 N2=N3-N1;
528   2                                 N4=N3*3;
529   2                                 N5=N4+N1;
530   2                                 N6=N5+N1;
531   2                                 POS=FLOAT(SIGNED(N1));
532   2                                 ALPHA=1./(18.*POS*POS*POS);
533   2                                 POS=0.;VIT=POS;ACC=POS;
536   2                                 DO I=1 TO NN;
537   3                                 IF I<=N1 THEN DO; ACC=ACC+1.;GOTO ETQ; END;
542   3                                 IF I<=N2 THEN GOTO ETQ;
544   3                                 IF I<=N3 THEN DO; ACC=ACC-1.;GOTO ETQ; END;
549   3                                 IF I<=N4 THEN GOTO ETQ;
551   3                                 IF I<=N5 THEN DO; ACC=ACC-1.; GOTO ETQ; END;
556   3                                 IF I<=N6 THEN GOTO ETQ;
558   3                                 ACC=ACC+1.;
559   3                                 ETQ: VIT=VIT+ACC;
560   3                                      POS=POS+ALPHA*VIT;
561   3                                      DO J=1 TO 6;
562   4                                      W(J)=TI(J)+POS*DEL(J);
563   4                                      END;
564   3                                 CALL INTER;
565   3                                 IN=6;CALL TRANSF(@IN,@W(0));
567   3                                 IF IN=1 THEN RETURN;
569   3                                 CALL COUPD;
570   3                                 END;
571   2                                 END FILTR;
          /*****************************************************************/
          /*                          DEPLG                                */
          /*****************************************************************/
572   1                               DEPLG: PROC;
573   2                                 DCL I BYTE;
574   2                                 DCL NTIM WORD;
575   2                                 DO I=1 TO 6;
576   3                                 W(I+40)=FLOAT(SIGNED(TABARG(I)));
577   3                                 END;
578   2                                 DO I=4 TO 6;
579   3                                 W(I+40)=W(I+40)/RR(11);
580   3                                 END;
581   2                                 NTIM=TABARG(7);
```

```
 582   2                          CALL FILTR(NTIM);
 583   2                          END DEPLG;
 584   1                    DEPL: PROC;
 585   2                          DCL I BYTE;
 586   2                          DCL NTIM WORD;
 587   2                          DO I=1 TO 3;
 588   3                           W(I+40)=FLOAT(SIGNED(TABARG(I)));
 589   3                           W(I+43)=W(I+3);
 590   3                          END;
 591   2                          NTIM=TABARG(4);
 592   2                          CALL FILTR(NTIM);
 593   2                          END DEPL;
 594   1                   DROTG: PROC;
 595   2                          DCL I BYTE;
 596   2                          DCL NTIM WORD;
 597   2                          DO I=1 TO 3;
 598   3                          W(I+40)=W(I);W(I+43)=FLOAT(SIGNED(TABARG(I)))/RR(11);
PL/M-86 COMPILER     NTR23                                                    20/12/81

 600   3                          END;
 601   2                          NTIM=TABARG(4);
 602   2                          CALL FILTR(NTIM);
 603   2                          END DROTG;
 604   1                   LOGE:  PROC(K,BRG);
 605   2                          DCL (I,K) BYTE;
 606   2                          DCL NTIM WORD;
 607   2                          DCL BRG REAL;
 608   2                          NTIM=TABARG(2);
 609   2                          DO I=1 TO 6;
 610   3                            W(I+40)=W(I);
 611   3                          END;
 612   2                          I=K;
 613   2                          W(I)=W(I)+(FLOAT(SIGNED(TABARG(1)))/BRG);
 614   2                          CALL FILTR(NTIM);
 615   2                          END LOGE;
 616   1                      DROT1: PROC;
 617   2                          DCL I BYTE;
 618   2                          I=44;
 619   2                          CALL LOGE(I,RR(11));
 620   2                          END DROT1;
```

```
621   1                          DROT2: PROC;
622   2                          DCL I BYTE;
623   2                          I=45;
624   2                          CALL LOGE(I,RR(11));
625   2                          END DROT2;
626   1                         DROT3:PROC;
627   2                           DCL I BYTE;
628   2                           I=46;
629   2                          CALL LOGE(I,RR(11));
630   2                           END DROT3;
631   1                        DMPX: PROC;
632   2                           DCL NTIM WORD;
633   2                           DCL ARG REAL;
634   2                           ARG=FLOAT(SIGNED(TABARG(1)));
635   2                           NTIM=TABARG(2);
636   2                           CALL EPX(ARG);CALL ROTI;
638   2                           CALL FILTR(NTIM);
639   2                           END DMPX;
640   1                         DMPY:PROC;
641   2                           DCL NTIM WORD;
642   2                           DCL ARG REAL;
643   2                           ARG=FLOAT(SIGNED(TABARG(1)));
644   2                           NTIM=TABARG(2);
645   2                           CALL EPY(ARG);CALL ROTI;
647   2                           CALL FILTR(NTIM);
648   2                           END DMPY;
649   1                         DMPZ: PROC;
650   2                           DCL NTIM WORD;
651   2                           DCL ARG REAL;
652   2                           ARG=FLOAT(SIGNED(TABARG(1)));
653   2                           NTIM=TABARG(2);
654   2                           CALL EPZ(ARG);CALL ROTI;
656   2                           CALL FILTR(NTIM);
657   2                           END DMPZ;
658   1                       MPXYZ: PROC(X,Y,Z);
659   2                             DCL (X,Y,Z) POINTER;
PL/M-86 COMPILER    NTR23                                                    20/12/81
```

```
660   2                        DCL AX BASED X REAL;
661   2                        DCL AY BASED Y REAL;
662   2                        DCL AZ BASED Z REAL;
663   2                        W(1)=W(1)+AX*(W(8)*W(12)-W(11)*W(9))+AY*W(7)+AZ*W(10);
664   2                        W(2)=W(2)+AX*(W(10)*W(9)-W(7)*W(12))+AY*W(8)+AZ*W(11);
665   2                        W(3)=W(3)+AX*(W(7)*W(11)-W(10)*W(8))+AY*W(9)+AZ*W(12);
666   2                        END MPXYZ;
667   1                     MEPXYZ: PROC;
668   2                        DCL ARG(4) REAL;
669   2                        DCL I BYTE;
670   2                        DO I=1 TO 3;
671   3                        ARG(I)=(FLOAT(SIGNED(TABARG(I))))*0.5;
672   3                        END;
673   2                        CALL MPXYZ(@ARG(1),@ARG(2),@ARG(3));
674   2                        IN=6;CALL TRANSF(@IN,@W(0));
676   2                        IF IN=6 THEN DO;CALL COUPD;RETURN;END;
681   2                        DO I=1 TO 3;ARG(I)=-ARG(I);END;
684   2                        CALL MPXYZ(@ARG(1),@ARG(2),@ARG(3));
685   2                        END MEPXYZ;
686   1                  SAUV: PROC;
687   2                     DCL I BYTE;
688   2                     DO I=1 TO 12;W(I+20)=W(I);END;
691   2                     END SAUV;
692   1                  RESTOR: PROC;
693   2                     DCL I BYTE;
694   2                     DO I=1 TO 12;W(I)=W(I+20);END;
697   2                     END RESTOR;
698   1                  EVALX:PROC(IE1,IS1);
699   2                    DCL (IE1,IS1) POINTER;
700   2                    DCL IE BASED IE1 BYTE;
701   2                    DCL IS BASED IS1 BYTE;
702   2                    W(IS)=W(IE+1)*W(IE+5)-W(IE+4)*W(IE+2);
703   2                    W(IS+1)=W(IE+3)*W(IE+2)-W(IE)*W(IE+5);
704   2                    W(IS+2)=W(IE)*W(IE+4)-W(IE+1)*W(IE+3);
705   2                    END EVALX;
706   1                 ROTX: PROC(SS,CC);
707   2                     DCL (SS,CC) POINTER;
708   2                     DCL S BASED SS REAL;
709   2                     DCL C BASED CC REAL;
710   2                     DCL (I,II,III) BYTE;
```

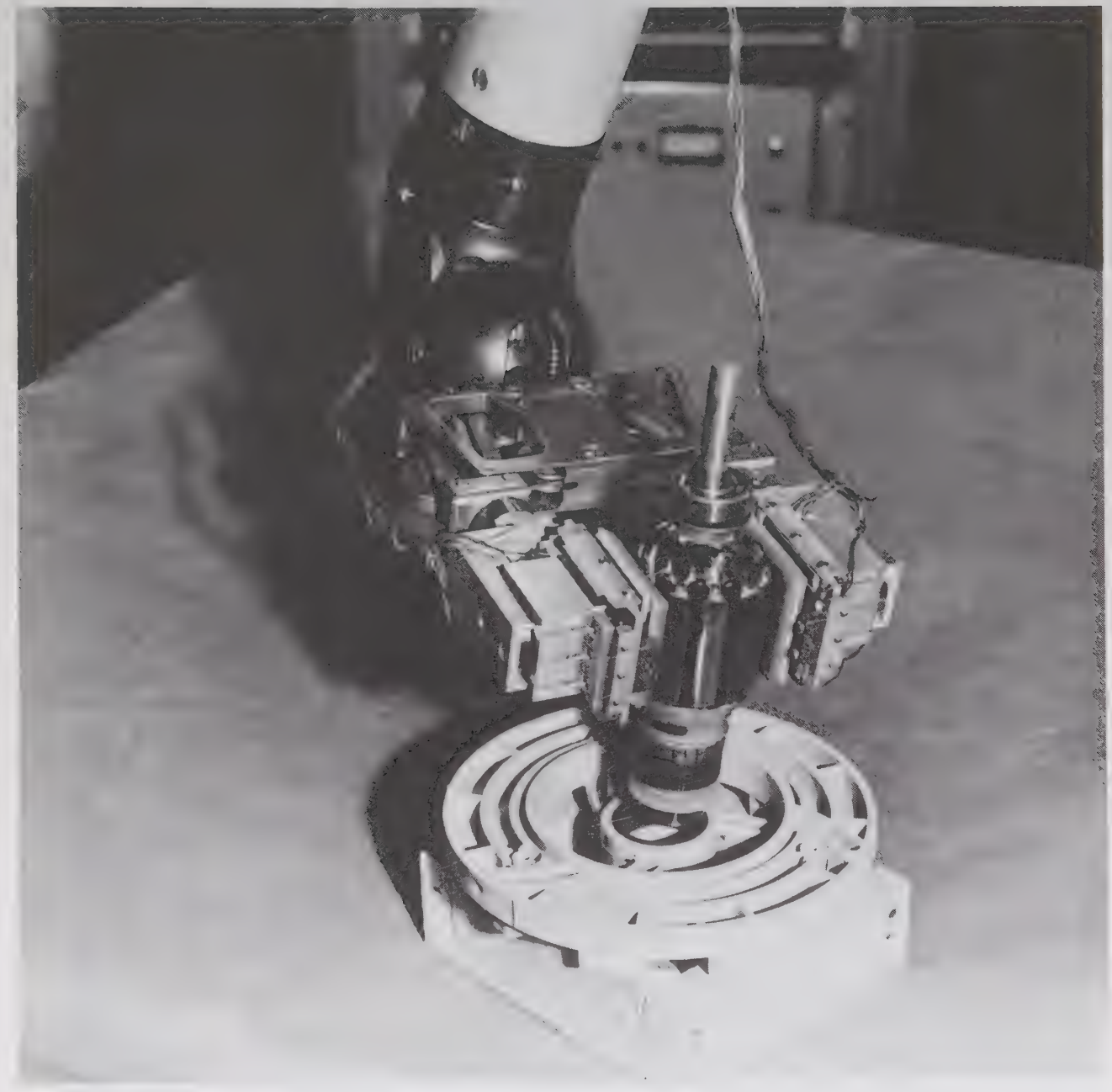
```
711   2                              DO I=7 TO 9;
712   3                              II=I+20;III=II+3;
714   3                              W(I)=C*W(II)+S*W(III);
715   3                              W(I+3)=C*W(III)-S*W(II);
716   3                              END;
717   2                              END ROTX;
718   1                           ROTY: PROC(SS,CC);
719   2                              DCL (SS,CC) POINTER;
720   2                              DCL S BASED SS REAL;
721   2                              DCL C BASED CC REAL;
722   2                              DCL (I,IE,IS) BYTE;
723   2                              IE=27;IS=33;CALL EVALX(@IE,@IS);
726   2                              DO I=10 TO 12;
727   3                              W(I)=S*W(I+23)+C*W(I+20);
728   3                              END;
729   2                              END ROTY;
730   1                           ROTZ: PROC(SS,CC);
PL/M-86 COMPILER    NTR23                                                    20/12/81

731   2                              DCL (SS,CC) POINTER;
732   2                              DCL S BASED SS REAL;
733   2                              DCL C BASED CC REAL;
734   2                              DCL (I,IE,IS) BYTE;
735   2                              IE=27;IS=33;CALL EVALX(@IE,@IS);
738   2                              DO I=7 TO 9;
739   3                              W(I)=C*W(I+20)-S*W(I+26);
740   3                              END;
741   2                              END ROTZ;
742   1                           REP:PROC;
743   2                              DCL (I,I1,I2,I3,I4,I5,I6) BYTE;
744   2                              W(49)=FLOAT(SIGNED(TABARG(1)));W(49)=W(49)*0.001;
746   2                              W(50)=RACS(W(49));CALL SAUV;
748   2                              DO CASE IN;
749   3                              DO;CALL ROTX(@W(49),@W(50));I1=2;I2=26;I3=6;
754   4                                 I4=3;I5=29;I6=9;I6=9;END;
759   3                              DO;CALL ROTY(@W(49),@W(50));I1=1;I2=32;I3=35;
764   4                                 I4=3;I5=29;I6=9;END;
768   3                              DO;CALL ROTZ(@W(49),@W(50));I1=1;I2=32;I3=35;
773   4                                 I4=2;I5=26;I6=6;END;
777   3                              END;
```

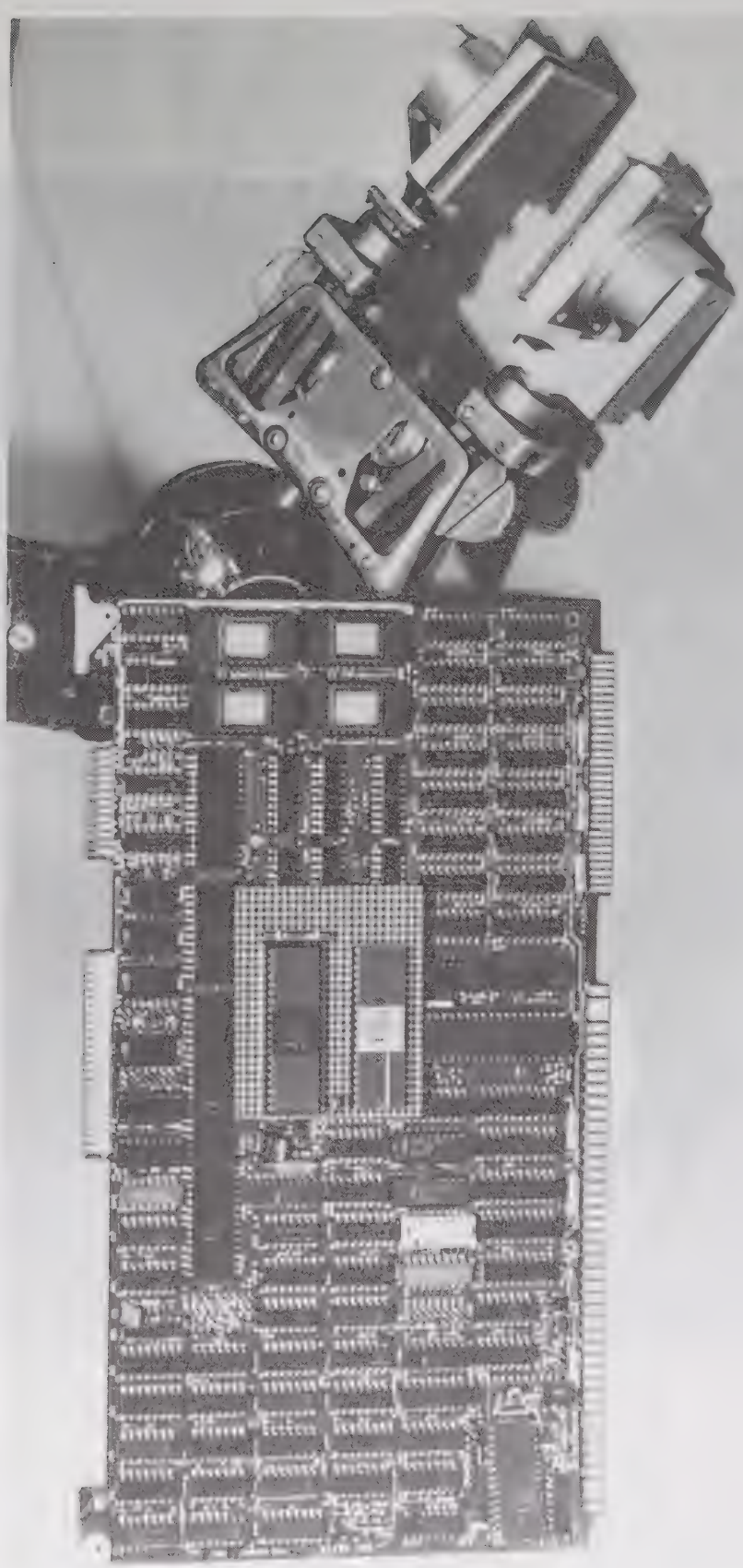
```
 778   2                          DO I=1 TO 3;
 779   3                          W(I)=W(I)+TR(I1)*(W(I+I2)-W(I+I3))+TR(I4)*(W(I+I5)-W(I+I6));
 780   3                           END;
 781   2                          IN=6;CALL TRANSF(@IN,@W(0));
 783   2                          IF IN<6 THEN DO;CALL RESTOR;RETURN;END;
 788   2                          CALL COUPD;
 789   2                          END REP;
 790   1                           DECROT:PROC;
 791   2                            DCL (I,IE,IS) BYTE;
 792   2                           CALL SAUV;
 793   2                            DO I=1 TO 3;
 794   3                            IE=I+3;W(IE)=FLOAT(SIGNED(TABARG(I)));W(IE)=W(IE)*0.001;
 797   3                            END;
 798   2                            CALL INTER;IE=7;IS=4;CALL EVALX(@IE,@IS);
 802   2                            DO I=1 TO 9;
 803   3                            ROT(I)=W(I+3);
 804   3                            END;
 805   2                            CALL RESTOR;
 806   2                            END DECROT;
 807   1                         MEP: PROC;
 808   2                            DCL (I,IE,IS) BYTE;
 809   2                            W(50)=FLOAT(SIGNED(TABARG(1)));CALL SAUV;
 811   2                            IE=27;IS=33;CALL EVALX(@IE,@IS);
 814   2                            IE=(IN-1)*3+1;
 815   2                            DO I=1 TO 3;
 816   3                            W(I)=W(I)+W(50)*(ROT(IE)*W(I+32)+
                                          ROT(IE+1)*W(I+26)+ROT(IE+2)*W(I+29));
 817   3                            END;
 818   2                            IN=6;CALL TRANSF(@IN,@W(0));
 820   2                            IF IN<6 THEN DO ;CALL RESTOR; RETURN;END;
 825   2                            CALL COUPD;
 826   2                            END MEP;
 827   1                            FERMA: PROC;
 828   2                           DCL (NTIM,I) WORD;
 829   2                           NTIM=TABARG(2);
 830   2                           W(20)=(80.-FLOAT(SIGNED(TABARG(1))))*12.8-W(19)+RR(7);
PL/M-86 COMPILER     NTR23                                                    20/12/81
```

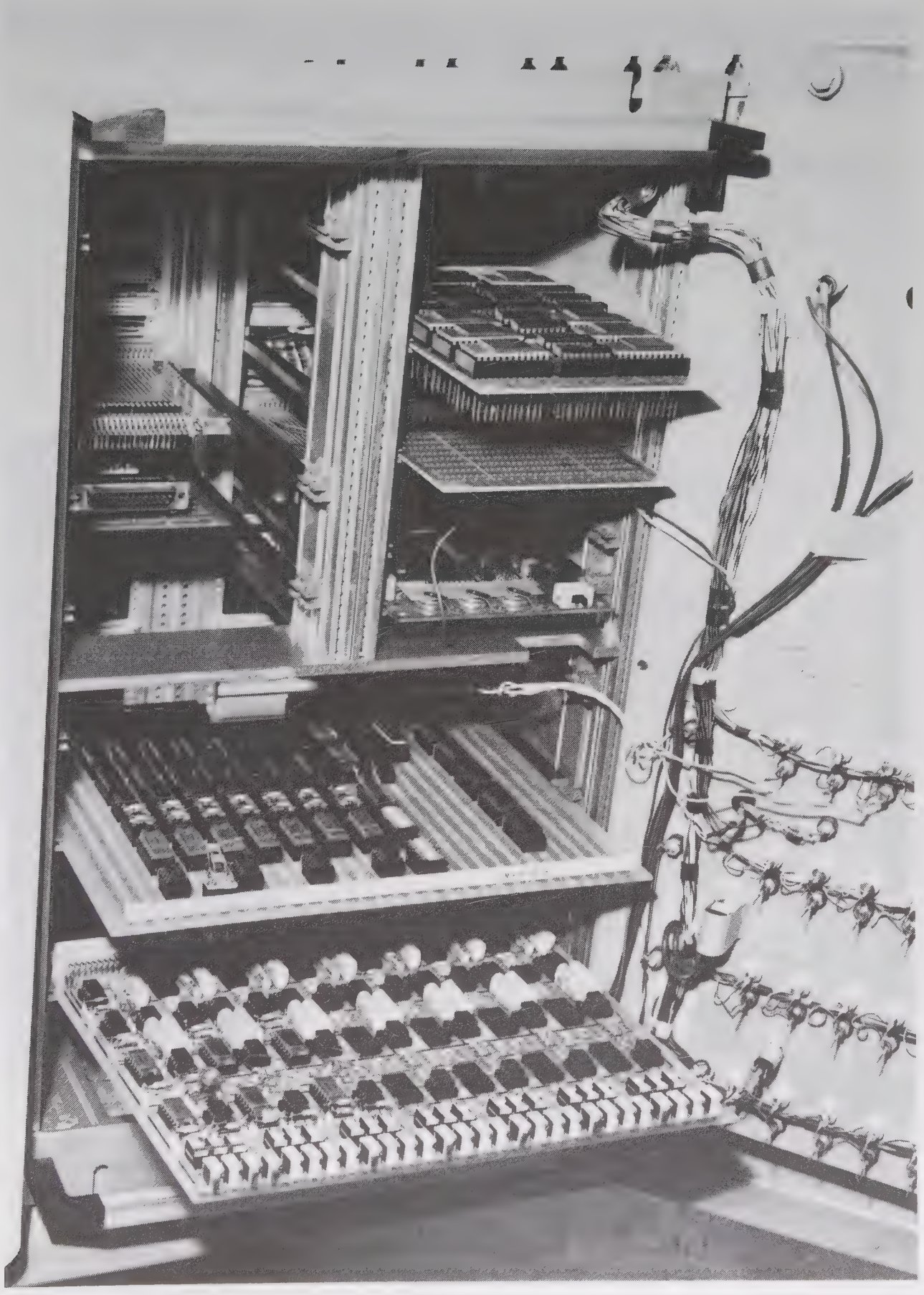
```
831   2                          W(20)=W(20)/FLOAT(SIGNED(NTIM));
832   2                          DO I=1 TO NTIM;
833   3                          W(19)=W(19)+W(20);TABARG(1)=UNSIGN(FIX(W(19)));
835   3                          CALL ENVCONS(40H);CALL DELAI(50);
837   3                          END;
838   2                          END FERMA;
      /*************************************************************************/
      /*                          PROGRAMME PRINCIPAL                          */
      /*************************************************************************/
                           /*  INITIALISATION DES PORTS */
839   1                          DISABLE;
840   1                          CALL INIT$REAL$MATH$UNIT;
841   1                          CALL SET$REAL$MODE(CONTROL$87);
842   1                          K=80H;
843   1                          OUTPUT(SMCD)=INI1;
844   1                          OUTPUT(EMCD)=INI3;
845   1                          OUTPUT(EPCD)=INI5;
846   1                          OUTPUT(SPCD)=INI7;
847   1                           DO I=1 TO 7;
848   2                           OUTPUT(SPA)=0F7H;
849   2                           OUTPUT(SPB)=0FFH;
850   2                           K=ROL(K,1);
851   2                           OUTPUT(SPC)=K;
852   2                           OUTPUT(SPC)=0;
853   2                           END;
                            /* ATTENTE DES IT              */
854   1                          OUTPUT(SMCD)=ARM1;
855   1                          OUTPUT(EMCD)=ARM2;
856   1                          OUTPUT(SMA)=READY;
                            /*   BOUCLE D'ATTENTE          */
857   1                          BOUCLE$C:DO WHILE (INPUT(EMC) AND MASK$ECRI)=0;
858   2                          END;
859   1                          INDEB:EN=INPUT(EMB)+256*INPUT(EMA);
860   1                          IF (EN AND MASKC)=0 THEN GO TO BOUCLE$C;
862   1                          DO I=1 TO (EN AND MASK$NBA);
863   2                          DO WHILE (INPUT(EMC) AND MASK$ECRI)=0;
864   3                          END;
865   2                          TABARG(I)=(INPUT(EMB)+256*INPUT(EMA));
866   2                          IF (TABARG(I) AND MASKC)<>0 THEN GO TO INDEB;
868   2                          TABARG(I)=TABARG(I)-16384;
869   2                          END;
```

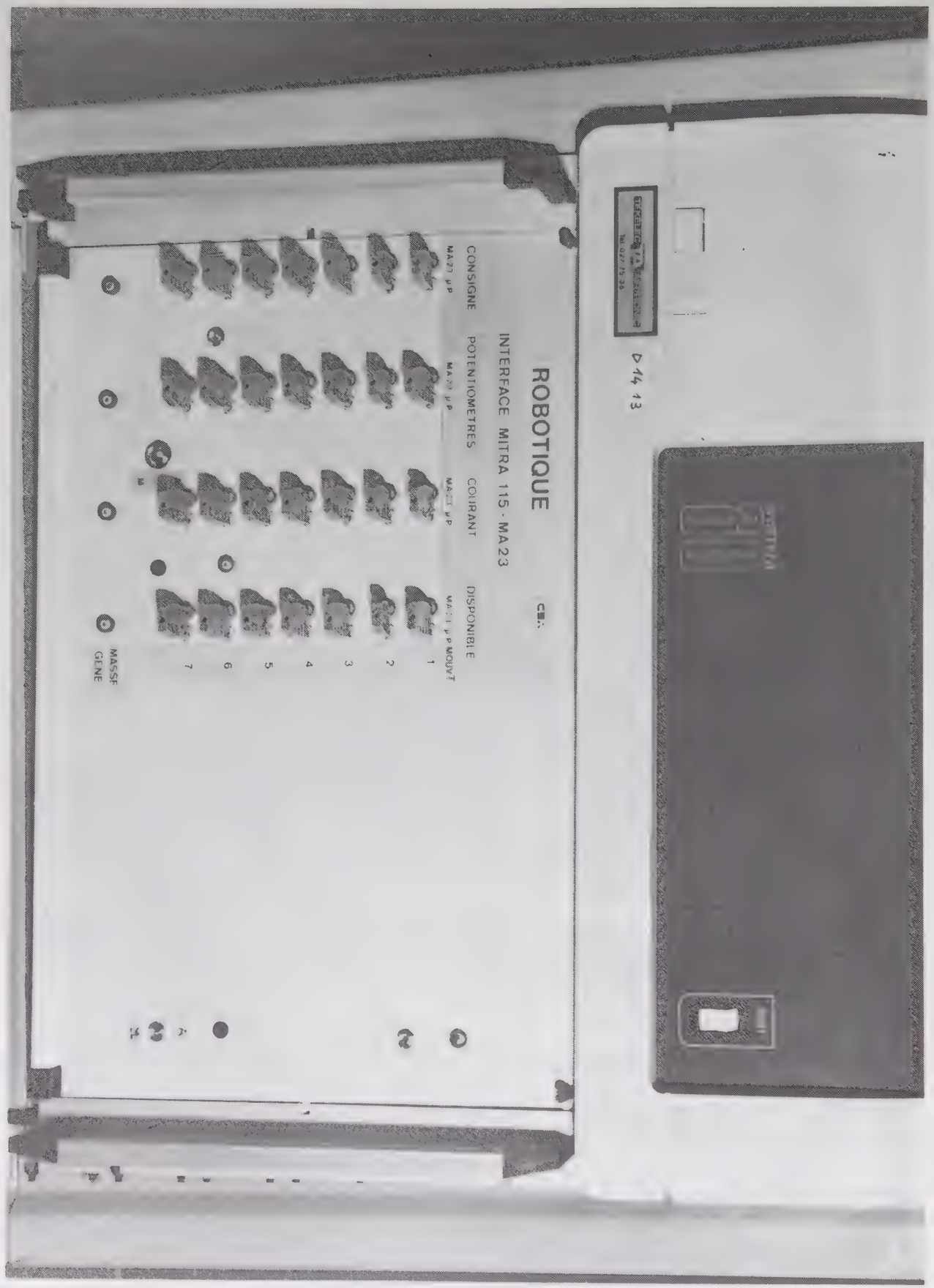
```
870   1                         DO CASE ((EN AND MASK$FON)/16);
871   2                         ;
872   2                         CALL ENVSIXCONS;
873   2                         CALL ENVCONS(40H);
874   2                         DO;N=6;ADR=0H;CALL LEC$POTARS(N,ADR);END;
879   2                         DO;N=1;ADR=6H;CALL LEC$POTARS(N,ADR);END;
884   2                         DO;N=6;ADR=8H;CALL LEC$POTARS(N,ADR);END;
889   2                         DO;N=1;ADR=10H;CALL LEC$POTARS(N,ADR);END;
894   2                         CALL LEC$POTARS(1,10H);
895   2                         CALL LEC$POTARS(1,11H);
896   2                         CALL LEC$POTARS(1,12H);
897   2                         CALL LEC$POTARS(1,13H);
898   2                         CALL LEC$POTARS(1,14H);
899   2                         CALL LEC$POTARS(1,15H);
900   2                         CALL LEC$POTARS(1,16H);
901   2                         ;
PL/M-86 COMPILER    NTR23                                                    20/12/81

902   2                         CALL DEPART;
903   2                         CALL FERMA;
904   2                         DO;IN=1;CALL MEP;END;
908   2                         DO;IN=2;CALL MEP;END;
912   2                         DO;IN=3;CALL MEP;END;
916   2                         DO;IN=0;CALL REP;END;
920   2                         DO;IN=1;CALL REP;END;
924   2                         DO;IN=2;CALL REP;END;
928   2                         DO;DO I=1 TO 3;TR(I)=FLOAT(SIGNED(TABARG(I)));END ;END;
933   2                         CALL DECROT;
934   2                         ;
935   2                         ;
936   2                         CALL DEPLG;
937   2                         CALL DEPL;
938   2                         CALL DROTG;
939   2                         ;
940   2                         ;
941   2                         ;
942   2                         CALL DROT1;
943   2                         CALL DROT2;
944   2                         CALL DROT3;
945   2                         CALL DMPX;
```

```
 946   2                          CALL DMPY;
 947   2                          CALL DMPZ;
 948   2                          CALL MEPXYZ;
 949   2                          ;
 950   2                          ;
 951   2                          ;
 952   2                          ;
 953   2                          END;
 954   1                          OUTPUT(SMA)=READY;
 955   1                          GO TO BOUCLE$C;
 956   1                          END NTR23;


MODULE INFORMATION:

     CODE AREA SIZE     = 2C5CH  11356D
     CONSTANT AREA SIZE = 0000H      0D
     VARIABLE AREA SIZE = 02F9H    761D
     MAXIMUM STACK SIZE = 0032H     50D
     895 LINES READ
     0 PROGRAM ERROR(S)

END OF PL/M-86 COMPILATION
```








ROBOTIQUE
INTERFACE MITRA 115 - MA 23
CONSIGNE
POTENTIOMETRES
COURANT
DISPONIBLE
1
2
3
4
5
6
7
MASSE
GENE
D 14 13